Cranfield University School of Engineering
Automotive Studies Group

Doctor of Philosophy Thesis

Lap Time Simulation with Transient Vehicle and Tyre Dynamics

by

Daniel Patrick Kelly

7th May 2008

Supervisors: Prof R.S. Sharp, Prof M. Harrison, Prof N. Vaughan

Abstract

A numerical method is presented for the time optimal control of the race car. The method is then used to perform the role of the driver in numerical simulations of manoeuvres at the limit of race car performance. The method does not attempt to model the driver but rather replaces the driver with methods normally associated with numerical optimal control. The use of constraints on the method is then considered to represent the performance limits of the human driver. The method simultaneously finds the optimal driven line and the driver control inputs (steer, throttle and brake) to drive this line in minimum time. The method is in principle capable of operation with arbitrarily complex vehicle models as it requires only limited access to the vehicle model state vector. It also requires solution of the differential equation representing the vehicle model in only the forward time direction and is hence capable of simulating the full vehicle transient response. The impact of various vehicle parameters on minimum manoeuvre time, driven line and vehicle stability is shown for a number of representative manoeuvres using a quasi-steady state vehicle model. A similar process is then carried out to analyse the effect of suspension springs and dampers using a fully dynamic sprung vehicle model. The presented transient time optimal control method is then compared with results obtained from a traditional quasi-steady state manoeuvre time simulation method. A thermodynamic tyre model is developed and the time optimal control algorithm is used to evaluate dynamic tyre temperature effects on lap time and vehicle stability.

Contents

Acknowledgements

I would like to give special thanks to Robin Sharp for offering me the chance to carry out this work at Cranfield and his continuing interest in the work following his move to Imperial College. I would also like to thank my second supervisor Matthew Harrison for his support and in particular his understanding of the challenges faced by a part-time student. Thankyou also to Nick Vaughan for helping me through the final lap. I would also like to thank Daniele Casanova for his input during the early stages of this work and Pat Symonds for allowing me to use data and models from Daniele's work [8].

The following people have shown an interest in the project and have provided help, support and encouragement : Max Nightingale, Jonathan Tempest, Steve Wise, Martin Shaw, James Knapton, Ernesto Fina, Jonathan Ruffley.

List of Symbols

Abbreviation	Description	Definition
α	Tyre Slip angle	
α_{LF}	Slip angle at left front wheel	page 182
α_{RF}	Slip angle at right front wheel	page 182
α_{LR}	Slip angle at left rear wheel	page 182
α_{RR}	Slip angle at right rear wheel	page 182
A_y	Vehicle Lateral Acceleration	
β	Vehicle body slip angle	page 27
c_j	j^{th} Constraint value in an NLP problem	
C_d	Displacement (track boundary) constraint	page 25
C_{F_z}	Wheel load constraint	page 27
C_β	Body side slip (stability) constraint	page 27
C_κ	Tyre Slip Ratio (stability) constraint	page 26
C_{uang}	Understeer angle (stability) constraint	page 27
C_{heave}	Axle heave damping due to suspension	page 122
C_{roll}	Axle roll damping due to suspension	page 122
C_{tyre}	Tyre Vertical damping	page 123
$\bar{g}$	Gradient vector of objective function in an NLP problem	
D_{aero}	Fraction of total aerodynamic downforce on front axle (Aero balance)	page 183
D_{brake}	Fraction of total brake torque on front axle (Brake Balance)	page 184
$D_{heave_{spring}}$	Ratio of front to total heave stiffness due to vehicle suspension	page 124
D_{roll}	Roll Moment Distribution (RMD) or Roll Balance. Fraction of total lateral load transfer taken up by front axle	page 188
$D_{roll_{cp}}$	Roll Moment Distribution (RMD) or Roll Balance measured at vehicle contact patch	page 124
$D_{roll_{spring}}$	Ratio of front to total roll stiffness due to vehicle suspension	page 124
d_x, d_y	Displacement of vehicle centre of gravity in the plane	

Abbreviation	Description	Definition
δ	Steer angle	
$F_{x_{LF}}$	Left front tyre longitudinal force	
$F_{x_{RF}}$	Right front tyre longitudinal force	
$F_{x_{LR}}$	Left rear tyre longitudinal force	
$F_{x_{RR}}$	Right rear tyre longitudinal force	
$F_{y_{LF}}$	Left front tyre lateral force	
$F_{y_{RF}}$	Right front tyre lateral force	
$F_{y_{LR}}$	Left rear tyre lateral force	
$F_{y_{RR}}$	Right rear tyre lateral force	
$F_{z_{LF}}$	Left front tyre vertical force	
$F_{z_{RF}}$	Right front tyre vertical force	
$F_{z_{LR}}$	Left rear tyre vertical force	
$F_{z_{RR}}$	Right rear tyre vertical force	
$F_{zv_{LF}}$	Virtual Left front tyre vertical force	
$F_{zv_{RF}}$	Virtual Right front tyre vertical force	
$F_{zv_{LR}}$	Virtual Left rear tyre vertical force	
$F_{zv_{RR}}$	Virtual Right rear tyre vertical force	
h_F	Front roll centre height	page 121
h_R	Rear roll centre height	page 121
$I_z z$	Rotational inertia of vehicle about z axis (yaw inertia)	
J_d	Rotational inertia of engine and driveline (excluding wheels) referred to engine	
J_F	Rotational inertia of front wheel and tyre	
J_R	Rotational inertia of rear wheel and tyre	
κ	Tyre Slip ratio	
κ^{Lim}	Maximum allowable slip ratio at solution	
κ_{LF}	Slip ratio at left front wheel	page 183
κ_{RF}	Slip ratio at right front wheel	page 183
κ_{LR}	Slip ratio at left rear wheel	page 183
κ_{RR}	Slip ratio at right rear wheel	page 183
K_{heave}	Axle heave stiffness	page 122
K_{roll}	Axle roll stiffness due to suspension	page 122
$K_{roll_{cp}}$	Axle roll stiffness measured at tyre contact patch due to suspension and tyres	page 122
K_{tyre}	Tyre Vertical stiffness	page 123
L	Vehicle wheelbase	
L_s	Vehicle suspension link length	
μ_0	Tyre static friction	page 143
μ_d	Tyre dynamic friction	page 143
N_β	Yaw Stiffness Derivative	page 217
N_δ	Control Moment Derivative	page 218
N_r	Yaw damping Derivative	page 217
P	'Driver' combined throttle / brake pedal control input $\in [-1, 1]$	page 23
r	Vehicle yaw velocity about z axis	
R_F	radius of front tyre	

Abbreviation	Description	Definition
R_R	radius of rear tyre	
τ	Total gear ratio between engine and rear wheels	page 185
θ_{uang}	Vehicle understeer angle	
θ_p	Vehicle chassis pitch angle	page 181
θ_r	Vehicle chassis roll angle	page 181
θ_y	Vehicle chassis yaw angle (absolute rotation angle about z axis)	page 181
$\theta_{LF}, \theta_{RF}, \theta_{LR}, \theta_{RR}$	Angle between lower suspension links and chassis	page 121
θ_{heave}	Sum of left and right suspension angle, defines axle heave	page 121
θ_{roll}	Difference of left and right suspension angle, defines axle roll	page 121
v_x	Longitudinal Velocity at vehicle centre of gravity	
v_y	Lateral Velocity at vehicle centre of gravity	
$\bar{m}$	State vector of vehicle model	page 22
$\bar{u}$	vector of vehicle control inputs	page 11
W	Race track width	
$\bar{x}$	Vector of independent variables in an NLP problem	
W_F	Vehicle front axle track	
W_R	Vehicle rear axle track	

List of terms and Abbreviations

Abbreviation	Description	Definition
DAE	Differential Algebraic Equation	
MTM	Minimum time manoeuvring	
NLP	Non-Linear-Programming	
ODE	Ordinary Differential Equation	
QSS	Quasi-Steady-State	
RMD	Roll moment distribution	
SQP	Sequential Quadratic Programming	

Chapter 1

Introduction

Motor racing is a sickness and the only way you can get over it is to die

Ken Tyrrell 1924-2001

1.1 Introduction

The general problem in motor racing is to design and set-up a vehicle so as to allow it to complete a given manoeuvre or set of manoeuvres in the minimum possible time. It is common practice in the motor racing industry to construct mathematical models of racing vehicles and use numerical optimization methods to estimate minimum manoeuvre time. This is commonly referred to as lap time simulation. It will be referred to here as the minimum time manoeuvring (MTM) problem. The principal challenge faced by these methods are to emulate those functions normally performed by the driver. The main purpose of this work is to develop methods capable of carrying out this function. It is not the intention to model the actual processes carried out by a human driver but rather to replace the driver with techniques usually associated with optimal control problems. Constraints on vehicle state can be added to emulate driver performance limits. Milliken [36] reports that manoeuvre time simulation is believed to date back to Mercedes Benz in 1954-55 and possibly as far back as 1937-1939. In [36] Milliken describes the development of a series of computer codes for Quasi-steady-state (QSS) lap time simulation beginning in 1982. Any manoeuvre time simulation code will face a compromise between accuracy and speed of computation. Lap time simulators are typically used at trackside by engineers making vehicle set-up choices during a racing event. In a practical trackside code, the time to compute a solution must be counted in minutes and not hours. Typically, QSS simulators require signficantly less computer time than current transient methods in order to reach useful solutions. For this reason, the published work in the last century relates mainly to QSS methods. Around the turn of the century the power of the readily available personal

computer had reached the point where numerical optimal control methods could be used with transient vehicle models.

1.1.1 QSS manoeuvre time simulation

A detailed description of a quasi-steady-state manoeuvre time simulation is provided in Appendix D and in [4]. We summarise here the limitations of the QSS method in order to explain the motivation for the development of more sophisticated transient methods.

- Throughout the solution, the vehicle is at all times in a steady state trim with no unbalanced forces or moments. Vehicle transient behaviour such as 'turn in' cannot be simulated. Operation of the vehicle over bumps and kerbs on the road surface cannot be simulated.

- Suspension elements are at all times stationary with respect to the vehicle chassis. Hence suspension dampers cannot be simulated.

- The driven line is pre-specified hence changes in ideal line with vehicle set-up cannot be studied.

- The QSS method can be considered as a series of boundary value problems with the boundary conditions consisting of the vehicle state at the apex of each corner. QSS methods approach this problem by operating in both the forward and reverse time direction from each boundary value. Since it is necessary to pre-specify the vehicle state at possibly several boundary points along the manoeuvre, it is difficult to solve an initial value problem with a QSS method. This makes it difficult to model features such as a tyre model with thermal effects. Since in this case, it would be necessary to pre-specify the thermal state of the tyre at several points along the manoeuvre.

Despite the fact that it is quite a mature technology, there is very little published in the area of QSS manoeuvre time simulation. A recognised standard approach to the problem has not been published. Several commerical software packages are available with inbuilt QSS simulators. The general level of secrecy in the motor racing industry and the commercial nature of the available codes goes some way to explain the paucity of literature. At the highest levels of motorsport, teams use in house developed codes. Appendix D discusses this further.

1.1.2 Transient manoeuvre time simulation

The task of the racing driver can be viewed as one of constrained time optimal control. The driver must complete a given manoeuvre in minimum time, subject to a number of constraints. The constraints consist of:

- The spatial constraints that define the manoeuvre (the track boundaries)

- The performance limits of the car

The process by which the driver achieves this can be viewed as one of two stages [44]:

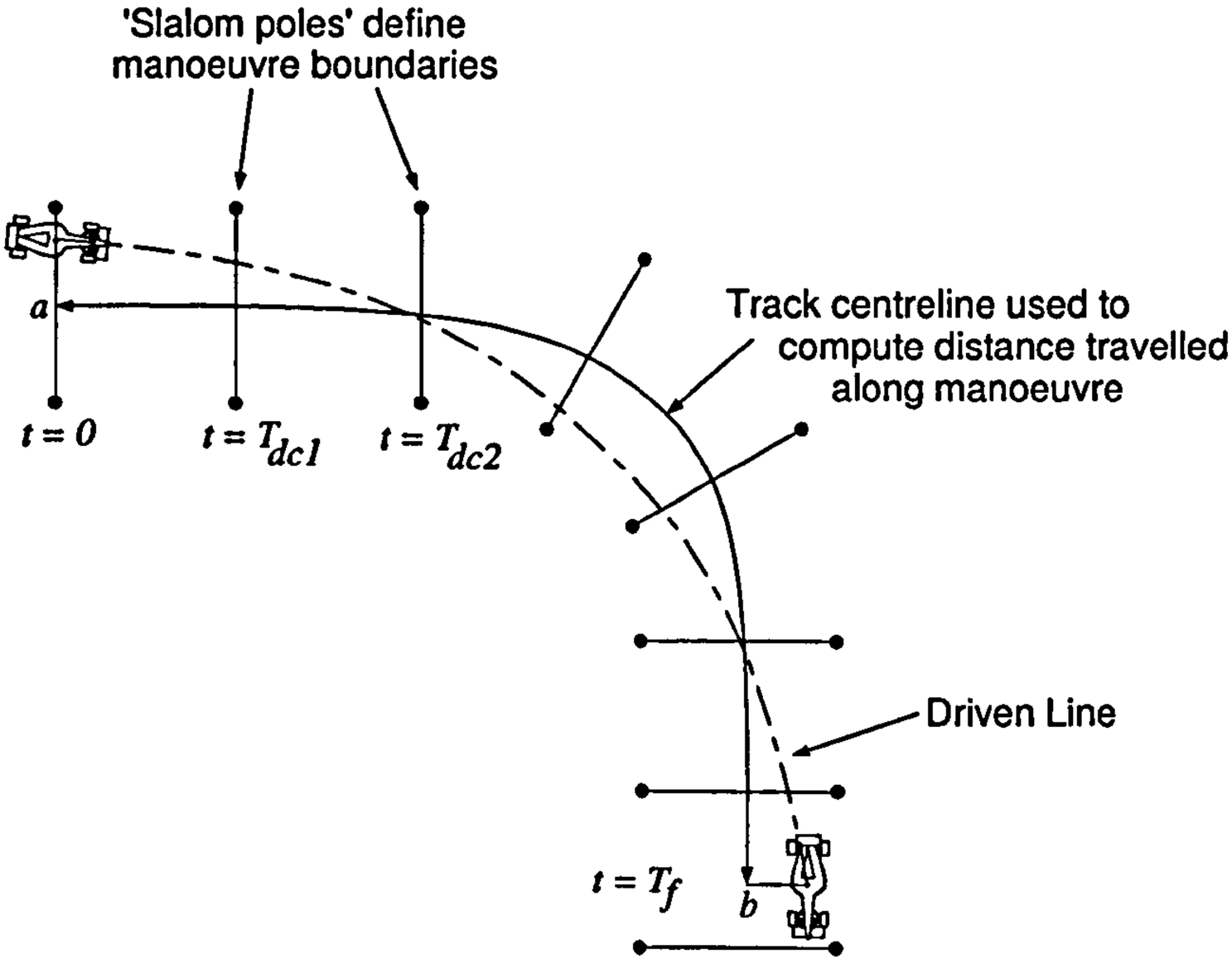


Figure 1.1: Example objective and constraints

- Plan a trajectory (The racing line and speed profile)

- Produce the necessary car controls to drive the car along that trajectory.

A transient MTM method must achieve all of the above using the full equations of motion of a vehicle model. The transient methods described in the literature are based on techniques developed for aerospace applications. Bryson [32] and Betts [50] describe a number of such applications. In these applications, the objective is to determine an optimal state trajectory that manoeuvres a non-linear vehicle model between two states. The trajectory is optimal in the sense that it minimizes some parameter such as energy used or time elapsed. Typically, the control inputs to the vehicle model are discretized over time and an interpolation scheme is used to render a continuous representation. The control values at the discrete points in time then become the indepedendent variables in a non-linear-programming (NLP) problem. An example formulation of the racing car minimum time manoeuvring (MTM) problem is given below. In this formulation of the problem, we simultaneously find the racing line and the controls needed to drive the line in minimum time. A vehicle model is typically described by a differential equation of the form:

$$\frac{d(\bar{m}(t))}{dt} = \bar{V}(\bar{m}(t), \bar{u}(t)) \tag{1.1}$$

Where $\bar{m}(t) \in R^{n_m}$ is the n_m dimensional state vector and $\bar{u}(t) \in R^{n_u}$ is the n_u dimensional control vector at time t. $\bar{V}$ is a vector valued function that computes the derivatives of the vehicle state vector. In essence, it describes the model. If we consider a manoeuvre on a time interval $[0, T_f]$, our performance metric can be defined so as to maximise the

distance travelled along a precscibed section of race track. This is a function of the final state $m(T_f)$. Figure 1.1 shows an example manoeuvre. The manoeuvre shape is specified by a curve defining the track centreline. The distance travelled by the vehicle at T_f is the length of this curve between the points marked a and b. Let $D(m(0), m(T_f))$ be a function that defines this length. The 'slalom' poles are used to constrain the vehicle path. A function is needed to define these constraints in a form suitable for NLP. For the i^{th} constraint the vehicle state as the vehicle crosses the line connecting a pair of poles at time T_{dci} is $m(T_{dci})$. Let $C_{di}(m(T_{dci})$ be a function that takes a value ≤ 0 if the vehicle is within the n^{th} pair of poles and > 0 if it is outside. The continuous time control vector $\bar{u}(t)$ is created from a discrete control vector $\bar{u}(k)...k = 1, 2, ...K$ by a suitable interpolation scheme. The MTM problem is that of finding suitable values for $\bar{u}(k)$ to maximise the distance travelled along the track whilst satisfying the displacement constraints. In summary:

Find:

$$\bar{u}(k)...k = 1, 2, ..K \tag{1.2}$$

to maximise:

$$D(m(0), m(T_f)) \tag{1.3}$$

subject to

$$C_{di}(m(T_{dci})) \leq 0...i = 1, 2, ...I \tag{1.4}$$

In order to make use of a general purpose non-linear-programming code, it is necessary to formulate the problem as a general NLP problem as shown below. It is assumed that the reader is familiar with NLP and a review is provided in Appendix C.

Maximize (or Minimize):

$$f(\bar{x}) \tag{1.5}$$

$$\bar{x} \in R^{n_x} \tag{1.6}$$

Subject to:

$$c_j(\bar{x}) \leq 0...j = 1...n \tag{1.7}$$

$$c_j(\bar{x}) = 0...j = n + 1...m \tag{1.8}$$

Where $f(\bar{x})$ is the scalar valued objective function and $\bar{x}$ is the n_x dimensional vector of independent variables. The scalar valued functions $c_j(\bar{x})$ compute the constraints. There are n inequality constraints and $(m - n + 1)$ equality constraints. To formulate the MTM problem as an NLP problem, the procedure by which the objective function and constraints are computed is as follows:

1. Solve the differential equation given by Eqn 1.1 on the interval $[0, T_f]$ given the state at $t = 0$ ($\bar{m}(t_0)$)with a given trial set of discretized controls $\bar{u}(k)$. This is effectively 'driving' the car forwards in time for a period of T_f seconds. This is achieved by a numerical integration procedure e.g. Runge-Kutta.

2. Whilst solving the differential equation, detect when the vehicle crosses the lines connecting each pair of slalom poles. At these time instants, compute the displacement constraint functions C_{di}.

3. Finally, Compute the distance travelled along the track $D(m(0), m(T_f))$

Suitable mappings between the independent variables $\bar{x}$ and the discretized controls $\bar{u}(k)$ are described in the next chapter. The displacement constraints C_{di} can be mapped one-to-one as inequality constraints in the NLP problem. In this example MTM formulation, no equality constraints are required. This above formulation describes the basic features of the MTM method developed in this thesis. The full formulation is given in chapter 2.

In practice, NLP codes are designed to work with problems having 'smooth' non-linear objective and constraint functions. We show in chapter 3 that both the objective and constraint functions can be non-smooth. In chapter 2 we describe a series of constraints on vehicle state that are designed to prevent the NLP code from exploring regions of the objective and constraint functions that are highly non-linear. The most non-linear regions are encountered when the vehicle 'spins' or enters a highly understeer regime. This problem increases with the length of the manoeuvre to be optimized.

The resulting NLP problem can run into numerical difficulties as a result of the large difference in objective and constraint gradients between controls that are earlier and later in the manoeuvre. This problem also increases with the length of the manoeuvre. This issue and the non-linearity problem places a limit on the length of the manoeuvre that can practically be solved. In chapter 6 we use Finite Horizon control as a technique to extend the method to arbitrarily long manoeuvres.

1.2 Literature Review

Solutions to the minimum time manoeuvring problem based on discretization of the driver controls and formulation as an NLP problem have been published by several authors. Published works begin in the year 2000. Most of the work discussed below was published during the course of the work described in this thesis, so the review has been an ongoing process. MTM methods in the form described above have very high computational demands when compared to QSS methods. For this reason, much of the literature focusses on techniques that can be used to reduce the computational burden.

1.2.1 Cranfield University

The earliest published work in this area is due to Casanova [8],[13],[12]. The MTM method developed by Casanova is based on the principles outlined in section 1.1.2. The method is suitable for long manoeuvres, for example a typical motor racing circuit with a of length 5km. In order to tackle the non-linearity and gradient spread problems, Casanova breaks

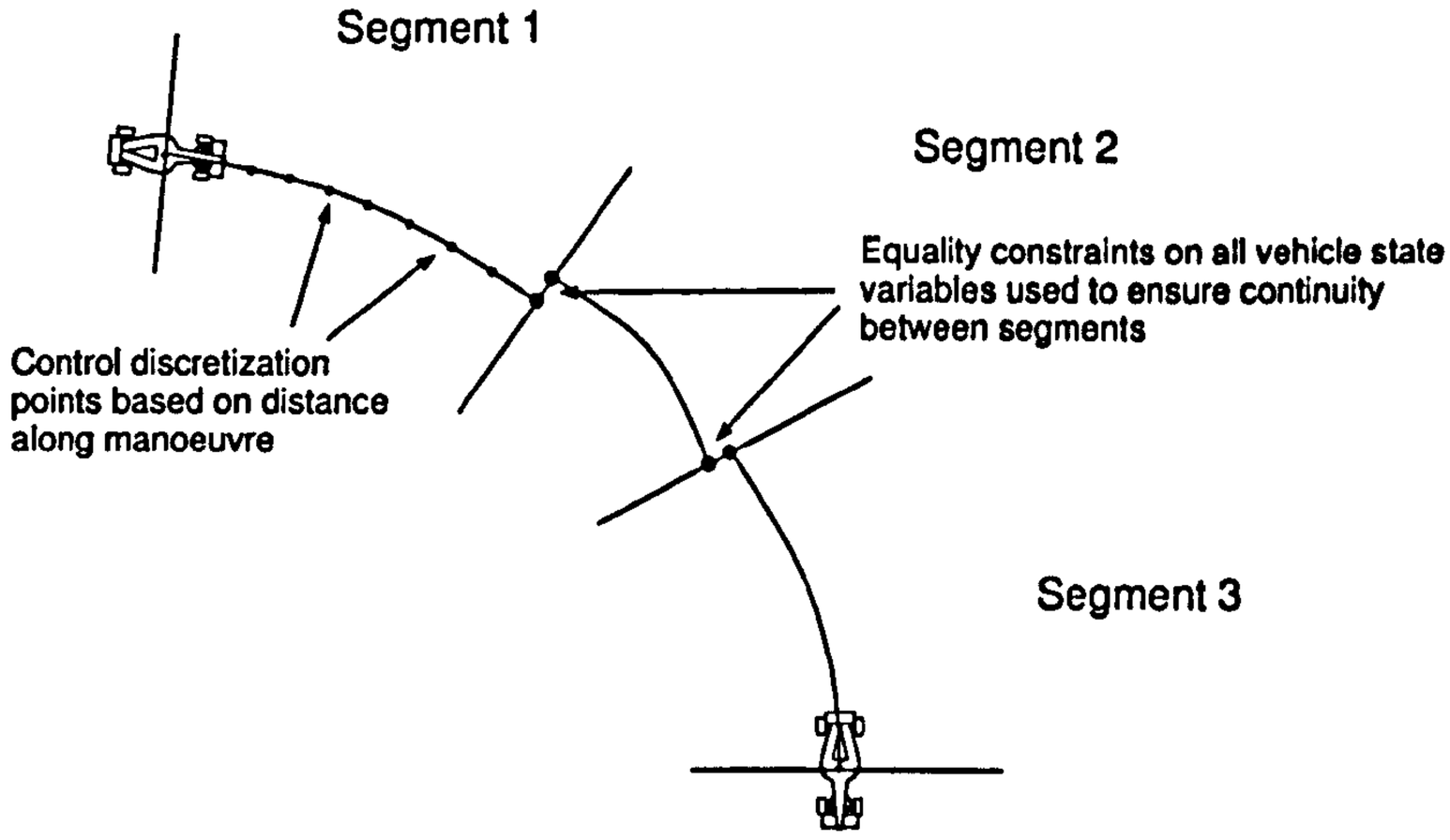


Figure 1.2: Method of Daniele Casanova

the manoeuvre into a series of segments (Figure 1.2)). A number of equality constraints are then added to the NLP problem to assure that the vehicle state at the end of a segment and the start of the next are equal. The vehicle model non-linearity problem is handled by reducing the length of the segments to the point where a 'spin' or 'severe understeer' does not have too long to develop before the end of the segment. The independent variable used in the solution of the vehicle differential equation and in the vehicle control generation functions is distance. A time-to-distance scaling factor is then applied in order to scale the vehicle state derivatives.

When solving a long manoeuvre, the number of independent variables and constraints in the NLP problem can be very large (> 1000). This places stringent demands on the fidelity of the objective and constraint gradients used by the NLP code. Automatic differentiation [28],[9] is used to compute the gradients. This is a technique that computes the gradients analytically by repeated application of the chain rule to the underlying model eqations used by the differntial equation solver. The code SNOPT [21] is used to solve the resulting NLP problem. Initial values for the controls are generated using a path and speed-profile-following algorithm and a pre-defined 'close' to optimal trajectory. Small random perturbations are applied to the resulting controls to generate multiple initial values for repeated trials. Casanova successfully uses the method to investigate the impact of vehicle longitudinal centre of mass location and yaw inertia on manoeuvre time.

Use of the vehicle state equality constraints and automatic differentiation requires that the MTM method have intimate knowledge of the vehicle model equations. The method developed in this thesis uses different methods in order to allow operation with a 'black box' vehicle model. The driver control input values used in this thesis are taken from the Casanova method. One of the vehicle models used in this work is very similar to the QSS model used by Casanova (see Appendix A).

1.2.2 Loughborough University

Best, Gordon [26],[25] and Komatsu [33] describe a number of interesting approaches to the MTM problem. In [26] a method is described that simultaneously optimizes the vehicle controls and a vehicle parameter (e.g. yaw moment of inertia). The driver controls consist of steer and torque. The discretized controls are rendered continuous by a zero order hold (no interpolation is used). The whole problem (including vehicle model) is formulated analytically. The Matlab symbolic toolbox [35] is then used to compute gradients. A steepest descent or conjugate gradient method are used to optimize the distance travelled in a given time. The method requires solution of vehicle equations in the reverse time direction. Results are presented analysing the effect of yaw moment of inertia on vehicle performance. In [33] a steering controller is formulated based on piecewise linearisation of a non-linear vehicle model. This results in a time varying linear system. Having rendered the system linear, an analytic optimal control method is used. This also requires solution of vehicle equations in the reverse time direction. In [25] a 'dual model' approach is described in order to reduce computational cost. The solution is found sequentially with a simpler model providing a reference velocity field for a more complex model. The authors state that the results produced are sub-optimal.

1.2.3 Georgia Institute of Technology, Atlanta

Velenis and Tsiotras have produced an extremely interesting series of papers [58][56][57][55]. Not all of this work describes solution of the MTM problem by NLP but the results presented are relevant to the general field of manoeuvre time simulation. The series begins [58] with a semi-analytical method to generate minimum-time optimal velocity profiles for a vehicle with given acceleration limits (e.g. a g-g diagram) driving along a specified path. The series continues [56] with a practical implementation of the semi-analytic method using receding horizon control. A speed profile is generated for the Silverstone Grand Prix circuit. In [57] the method is applied to a simple car model over a short manoeuvre. In [55] the method is used to compare minimum time and maximum exit velocity solutions.

1.2.4 University of Leeds

Siegler, Deakin and Crolla describe MTM methods in [16],[15]. In [15] they describe an MTM method suitable for short manoeuvres. The impact of objective and constraint non-linearity is not discussed. The driver control inputs are similar to the Casanova formulation [8]. Initial controls are generated with a path following algorithm with a reference driven line at a given constant speed. Finite differences are used to generate the objective and constraint gradients. The Matlab optimization toolbox [34] is used to solve the resulting NLP problem. A single corner manoeuvre is presented with path radius of approximately 17.5 m. The authors state that the optimized solution is not affected by the choice of constant velocity at the initial solution. The resulting optimized driven line is shown. In [16] a transient method is presented and the results compared with a QSS method. The transient method uses the throttle/brake values produced by a QSS method and a sinusoidal steering input. The authors conclude that the optimized manoeuvre time difference between transient and QSS methods is small.

1.3 Outline

We present below a brief summary of the thesis.

1.3.1 Chapter 2 - A Method for Short Manoeuvres

Chapter 2 introduces a transient method suitable for the solution of short manoeuvres. The method includes a number of novel features that support very robust operation in the presence of highly non-linear vehicle behaviour.

1.3.2 Chapter 3 - Performance of the MTM Algorithm

Chapter 3 provides some fundamental results using the method developed in chapter two and a QSS vehicle model (Appendix A). The accuracy and repeatability of the results is demonstrated. By performing many trials from randomly selected initial vehicle controls we demonstrate the consistency of the method and show that the MTM problem (as formulated in this work) possesses multiple locally optimal solutions. We consider the impact of the control discretization interval (e.g. controller sampling rate in Hz) on manoeuvre time. The optimal behaviour of the solutions is demonstrated by the use of tyre efficiency diagrams (Appendix B) and the six vehicle stability derivatives (Appendix A). We demonstrate that the vehicle yaw stiffness typically becomes negative in limit manoeuvring. It is also shown that yaw moment sensitivity to steer input falls to zero in limit manoeuvring. The MTM method is capable of extreme behaviour in its quest for minimum manoeuvre time. We consider additional constraints on the model to reflect limitations of human driver performance. The bandwidth of the human driver is considered.

1.3.3 Chapter 4 - Influence of vehicle parameters on Manoeuvre time, Driven line and Stability

For a number of manoeuvres, we establish the sensitivity of manoeuvre time, vehicle stability and driven line to a number of vehicle parameters. Vehicle mass, yaw moment of inertia, roll moment distribution, longitudinal centre of gravity and differential set-up are considered. Using a simple 90 degree corner followed by a straight, we show the impact on the manoeuvre time and final speed of the car on the 'look-ahead' distance along the straight. We also provide a comparison of minimum manoeuvre time and maximum exit speed cornering strategies.

1.3.4 Chapter 5 - Comparison of Transient and Steady State Solutions

Chapter 4 compares time optimal solutions obtained with the transient method with those obtained with a quasi-steady-state method. We analyse the effect of a change in racing line associated with a vehicle set-up change on the solutions produced by a quasi-steady-state simulator.

1.3.5 Chapter 6 - Extending the method for longer manoeuvres

Chapter 6 extends the method for longer manoeuvres using the method of finite horizon optimal control. We show the impact of distance to horizon on the driven line and optimal manoeuvre time. Results from a simulation of the Jerez circuit are presented.

1.3.6 Chapter 7 - Results using a Sprung Vehicle Model

We Provide some further results using a more complex vehicle model including sprung and unsprung masses. We consider vehicle operation when riding a serrated kerb. The effect of vehicle spring and damper rates on manoeuvre time and vehicle stability is considered.

1.3.7 Chapter 8 - Performance Analysis using a Thermodynamic Tyre Model

In many categories of motorsport, the thermal behaviour of the tyres is a dominant factor in the performance of the vehicle. In this chapter we develop from first principles a tyre model including thermal effects. The influence of track and ambient temperatures on manoeuvre time and vehicle stability is considered.

1.3.8 Chapter 9 - Conclusions

A summary of the main findings in the research is presented along with recommendations for further areas of work.

Chapter 2

A Method for Short Manoeuvres

Success can only be achieved with a kind of pioneer spirit and the repeated use of three tools: failure, introspection and courage

Sochiro Honda

2.1 Summary

In this chapter a numerical method is presented to solve the Minimum Time manoeuvring problem introduced in Chapter 1. It is a robust time optimal control algorithm for four wheeled vehicles subject to displacement constraints. The method simultaneously finds the optimal driven line and the controls required to follow the line in minimum time. Key features of the algorithm include:

- Capable of finding a solution from any initial control vector values. Hence does not require seeding with solution that is close to optimal. This allows a wide search of the solution space giving confidence that a global minimum has been located.

- Limited access to the state vector of the vehicle model is required. This allows the method to be used with arbitrarily complex vehicle models.

- Suitable for manoeuvres up to approximately 500m. The method is extended to arbitrarily long manoeuvres in chapter 6.

The problem is framed as a general non-linear-programming (NLP) problem. The NLP code CFSQP (C-Feasible Sequential Quadratic Programming) is then used to find the solution. The use of a feasible NLP method is key to the success of the algorithm. It is assumed that the reader is familiar with NLP and a summary is provided in appendix C.

2.2 Non-linear Programming

The method is formulated as a general NLP problem. The general NLP problem can be stated as follows.

Minimize:

$$f(\bar{x}) \tag{2.1}$$

$$\bar{x} \in R^{n_x} \tag{2.2}$$

Subject to:

$$c_j(\bar{x}) \leq 0 ... j = 1 ... n \tag{2.3}$$

$$c_j(\bar{x}) = 0 ... j = n+1 ... m \tag{2.4}$$

Where $f(\bar{x})$ is the scalar valued objective function and $\bar{x}$ is the n_x dimensional vector of independent variables. The scalar valued functions $c_j(\bar{x})$ compute the constraints. There are n inequality constraints (eqn 2.3) and $(m - n + 1)$ equality constraints (eqn 2.4). The problem can be also written so the goal is to maximise the objective function. In the form shown here the inequality constraints must be less than or equal to zero at the solution.

2.3 Time and Distance Based Formulations

In the general case of the MTM problem the value of the objective function is simply the time taken to complete a given manoeuvre. We will consider two formulations of the objective. In the first, it is to minimise the time taken to travel between a notional start and finish line on a section of race track. In the second, it is to maximise the distance travelled along a race track in a given fixed time. In each case, evaluation of the objective function consists of solving a differential equation initial value problem (representing the vehicle model). In both cases, time is used as the independent variable in the differential equation. Each formulation lends itself to a particular discretisation scheme for the driver control inputs and constraints. In the first formulation, we draw a number of lines across the race track perpendicular to the track centreline as shown in figure 2.1. These will be referred to as waylines. At any given time, the relative distance between the vehicle centre of gravity and the closest points on the preceding and following waylines forms a continuous variable as follows:

$$d_w = \frac{d_1}{d_1 + d_2} \tag{2.5}$$

This is used as the independent variable for the driver control input generation functions. The displacement constraints are specified at the time the vehicle centre of gravity crosses the wayline. The points where the wayline crosses the inside and outside of the track edge can be considered as slalom poles.

In the second formulation, time is used as the independent variable for both the driver controls and constraint functions. The objective is to maximise the distance travelled along

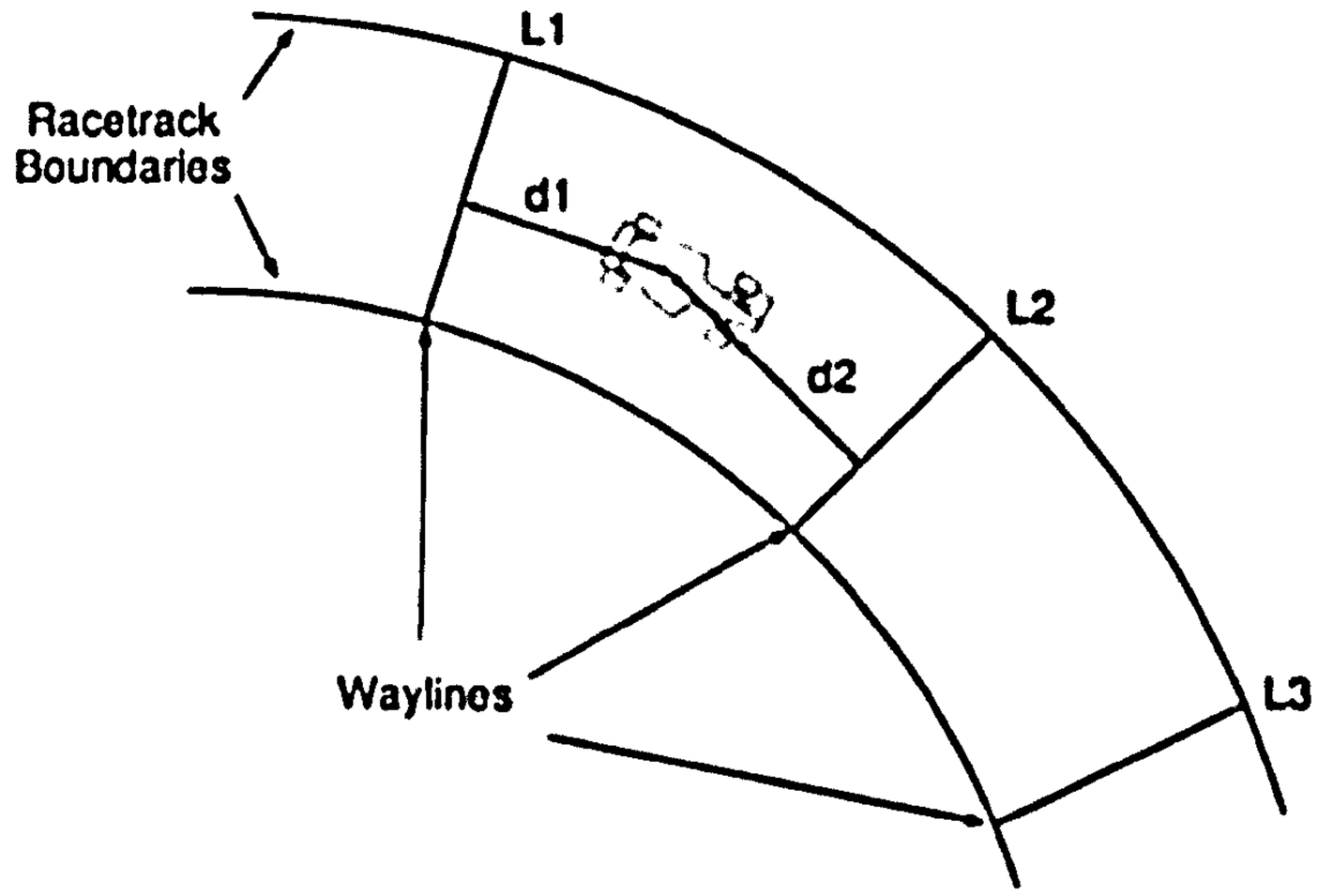


Figure 2.1: Distance based formulation

the race track in a given time. The method used to compute the distance travelled along the track and the distance of the vehicle from the centre of the track are shown in figure 2.2. The track centreline is discretized and then re-rendered as a continuous function using linear interpolation.

The motivation for using two formulations will become clear below. The distance based formulation is more convenient. The time based formulation is more robust.

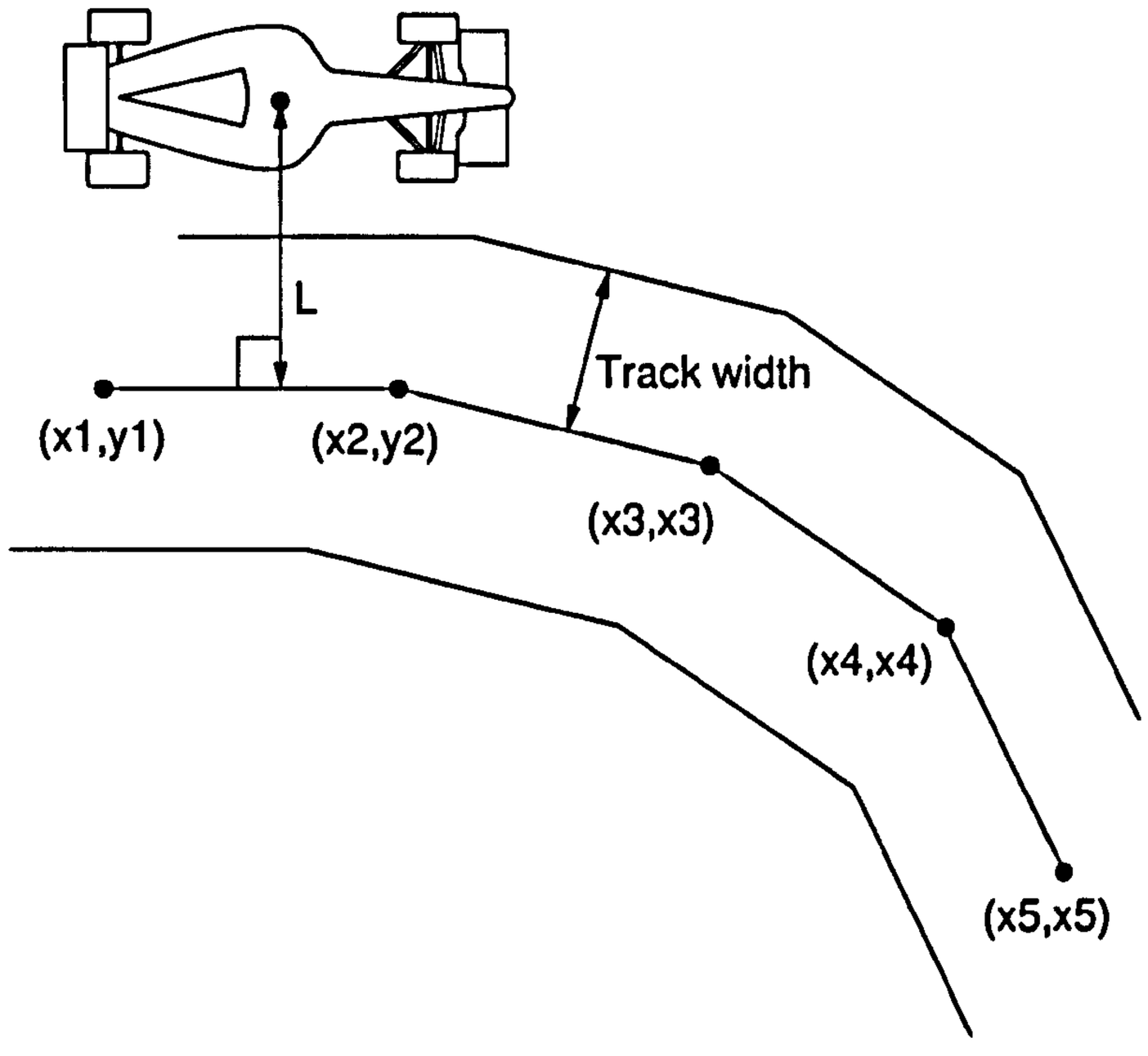


Figure 2.2: Vehicle Centre of Gravity Distance from track centre line

2.4 Vehicle Model

The vehicle model is described by the general differential equation:

$$\frac{d(\bar{m}(t))}{dt} = \bar{V}(\bar{m}(t), \bar{u}(t)) \tag{2.6}$$

Where $\bar{m}(t) \in R^{n_m}$ is the n_m dimensional state vector and $\bar{u}(t) \in R^{n_u}$ is the n_u dimensional control vector at time t. The dimension of the state vector is determined by the complexity of the vehicle model. The algorithm requires access to only those elements of the state vector that could generally be considered to be common to all vehicle models. The vehicle state vector elements required by the method are given in Table 2.1.

Symbol	Description
d_x, d_y	Displacement of vehicle centre of gravity in the plane
v_x	Longitudinal Velocity at vehicle centre of gravity
v_y	Lateral Velocity at vehicle centre of gravity
r	Vehicle yaw rate
A_y	Vehicle Lateral acceleration
κ_{LF}	Slip ratio at left front wheel
κ_{RF}	Slip ratio at right front wheel
κ_{LR}	Slip ratio at left rear wheel
κ_{RR}	Slip ratio at right rear wheel
$F_{z_{LF}}$	vertical load at left front wheel
$F_{z_{RF}}$	vertical load at right front wheel
$F_{z_{LR}}$	vertical load at left rear wheel
$F_{z_{RR}}$	vertical load at right rear wheel

Table 2.1: Vehicle State Vector values required by MTM Algorithm

2.5 Representation of the driver control inputs

The control inputs available to a driver in a typical vehicle model consist of engine torque demand, steering angle, braking torque demand, clutch torque transfer and gear ratio. For the purposes of this work this is simplifed to a 2 dimensional control vector consisting of a combined driving/braking torque and steering angle [8]. The independent variables used by the optimizer to represent the driving/braking torque are numbers in the range $[-1, 1]$. This can be thought of as two ranges. The range $[0, 1]$ represents the on-throttle regime. The value 1 represents maximum available engine driving torque. The value 0 represents the closed throttle scenario producing maximum engine braking. The range $[-1, 0[$ represents the braking regime. This is summarised in figure 2.3. The distribution of torques between axles and wheels including gearbox and differential is described in more detail in Appendix A. The steering controls simply represent the steer angle applied to the front wheels. Two schemes are described below which allow the vehicle controls to be represented at a number of discrete points. Interpolation is then used to produce the continuous signals for input to the vehicle model.

2.5.1 Distance Based Formulation

In this formulation, the optimizer independent variables specify the control values at the point where the vehicle crosses the wayline. At any time instant, the control value is computed based on d_w the relative distance from the centre of gravity of the vehicle and the nearest points on the two closest waylines as described in equation 2.5. If x_n and x_{n+1} represent the throttle/brake values at waylines n and $n+1$ then the throttle/brake value is given by:

$$u_T = x_n + (x_{n+1} - x_n) * d_w \tag{2.7}$$

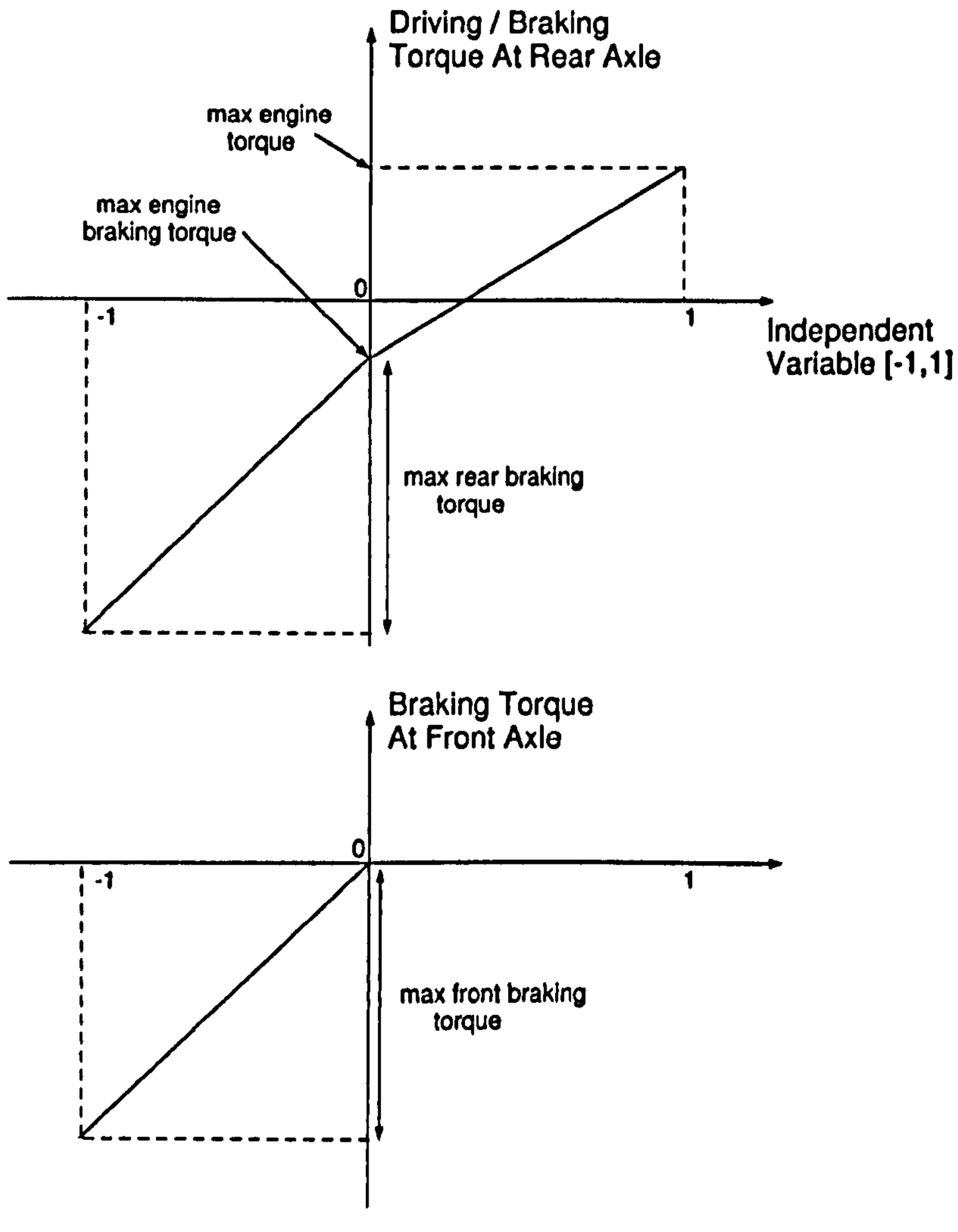


Figure 2.3: Driving/Braking torque representation

2.5.2 Time Based Formulation

In this fomulation, the optimizer independent variables specify the control values at discrete points in time. The continuous time control vector is then produced by linear interpolation over time.

2.6 Race track (displacement) constraints

Computation of the race track constraints is a two step process. Firstly we compute the distance of the vehicle centre of gravity from the race track centreline. Secondly we apply a function due to Casanova [8] to produce a function which is continuously differentiable (a requirement for SQP NLP codes). If the distance between the vehicle centre of gravity and the centre of the race track is d and the track width is W, then the displacement constraint is computed as shown in equation 2.8 and figure 2.4.

$$C_d = \left(\frac{2d}{W}\right)^2 - 1 \tag{2.8}$$

This provides a smooth function that is less than or equal to zero when the vehicle centre of gravity is within the track boundary and greater than zero otherwise. We choose to constrain only the vehicle centre of gravity to the racetrack for simplicity. This method could easily be extended to constrain the displacements of each of the four wheels. The approach described by equation 2.8 is used in other constraint formulations below to provide a smooth constraint function where it is desired to place some fixed limit on the value of a vehicle state variable.

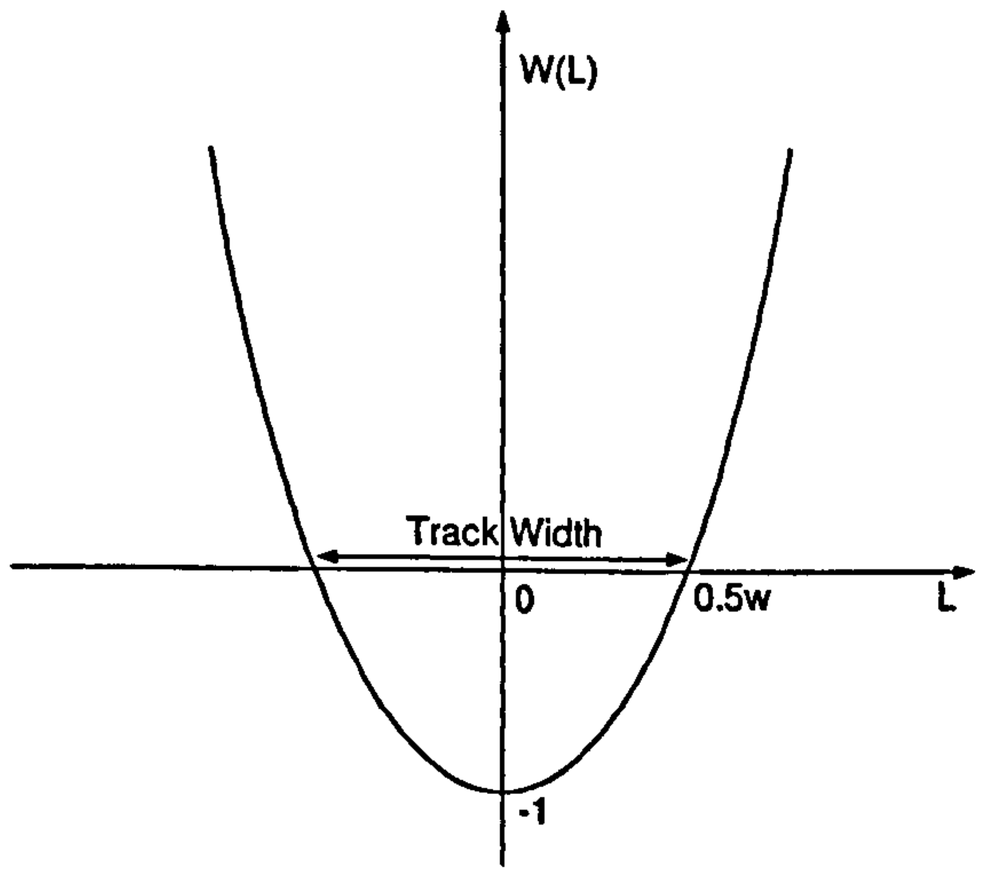


Figure 2.4: Differentiable displacement constraint function

2.6.1 Time Based Formulation

In this formulation the displacement constraints are specified at discrete points in time. At the specified instant, the distance of the vehicle centre of gravity from the track centre line is computed as shown in figure 2.2.

2.6.2 Distance Based Formulation

In this formulation the displacement constraints are specified at the time instant that the vehicle centre of gravity crosses the wayline. The distance of the vehicle centre of gravity from the centrepoint of the wayline is then computed.

2.7 Vehicle Stability constraints

The stability constraints are used to remove the most non-linear regions of the vehicle model from the space explored by the optimizer. This is a particularly important feature of the method that allows the optimizer to be seeded from randomly generated start points that are very far away from optimal solutions. We define a number of functions below that compute a stability parameter from the vehicle state vector. An individual stability constraint value is defined as the maximum value of such a function over a given time interval as described in equation 2.9

$$C_{stability} = max\left[C_f(s(t))\right] ...t \in [t_s, t_f[\tag{2.9}$$

In the distance based formulation t_s and t_f are defined as the time the vehicle crosses two waylines. As a result, the constraint is defined over a particular value of the racetrack. In the time based formulation t_s and t_f are specified at two fixed time intervals.

In the following chapters we show that the MTM algorithm is capable of exercising the vehicle to the extremes of its performance envelope in a manner that would not be possible by a human driver. The driver is limited by his own physiology and also by the requirement to leave some safety margin in the face of uncertainty and changing conditions such as tyre wear, changing track temperatures, wind direction, fuel load etc. The stability constraints also provide a means to limit the MTM algorithm to a more human like performance region. The constraints described below are those that are necessary to endow the MTM algorithm with robust performance. Further constraints are defined in chapter 3 in order to attempt to more closely match the performance of an expert human driver.

It should be noted that excluding some regions of the vehicle state vector $\bar{m}(t)$ from the optimizer search space does introduce the possibility that the region containing the globally optimal trajectory will be excluded.

2.7.1 Slip Ratios

In order to prevent the tyre from operating in highly super-saturated regions, we implement a primitive form of traction control and antilock braking. We place a limit on the absolute value of the slip ratio κ at the four wheels as described by 2.10.

$$C_\kappa = \left(\frac{\kappa}{\kappa^{Lim}}\right)^2 - 1 \tag{2.10}$$

2.7.2 Understeer angle

The understeer angle is a compact if crude way of describing the limit state of the vehicle. In unconstrained operation, it is typical for the optimizer to enter a highly understeer region of operation in which the steer control has very little effect. In order to exclude these regions we place a limit on the absolute value of the understeer angle (Appendix E as described by equation 2.11.

$$C_{uang} = \left(\frac{\theta_{uang}}{\theta_{uang}^{Lim}}\right)^2 - 1 \tag{2.11}$$

By default, this constraint also limits oversteer. In practice, oversteer usually results in a 'spin' which is then handled by the displacement and body side-slip constraints.

2.7.3 Body side-slip Angle

The vehicle body side-slip angle is the angle between the vehicle x axis and its vector velocity (equation 2.12).

$$\beta = tan^{-1}\frac{v_y}{v_x} \tag{2.12}$$

The linearity of the operating regime of the car can be described quite compactly by a phase plan plot of β and its derivative $\dot{\beta}$. Automotive stability control systems typically operate by limiting β [49]. In early generations of the manoeuvre time optimizer a constraint was defined which specified an approved region of the β / $\dot{\beta}$ phase plane. In later development it proved sufficient to place a simple limit on β. Equation 2.13 defines a primitive form of stability control. This is used to exclude high side-slip regions from the optimization process.

$$C_\beta = \left(\frac{\beta}{\beta^{Lim}}\right)^2 - 1 \tag{2.13}$$

2.7.4 Wheel loads

Vehicles exploiting aerodynamic downforce typically run very stiff suspensions. At limit performance, longitudinal and lateral load transfers are very high. This can result in one (usually front) wheel becoming fully unloaded during limit manoeuvring. We formulate the vehicle models so that two tyre loads are computed. The first load is representative of the real tyre physical load and cannot become negative. This is used in the model physics such as tyre force calculations. We also compute a virtual load that becomes negative as the tyre separates from the road surface. A constraint is then used to prevent this virtual load from becoming negative at the solution.

The F1 car has a tendency to fully unload a front tyre when accelerating out of a corner. Hence the wheel load constraints are typically active at MTM solutions. In implementing this constraint, we in effect limit the vehicle lateral load transfer to the value that would fully unload one wheel. It is acknowledged that this prevents the exploration of some of the vehicle state space but this is considered an acceptable compromise by the author. For each wheel, the constraint is described in equation 2.14 where F_z is the wheel load and F_z^{Lim} is the minimum permissible load for that wheel at the solution.

$$C_{Fz} = F_z^{Lim} - F_{zv} \tag{2.14}$$

2.8 Objective definition

The definition of the NLP objective function is described below for both the time and distance based formulations.

2.8.1 Time Based Formulation

In this formulation, the distance travelled by the car along a particular manoeuvre, in a given time interval $[t_s, t_f]$ is maximised. Computing the objectve function consists of solving an initial value differential equation problem (IVP) on the interval $[t_s, t_f]$ to find the vehicle state at time t_f. Then the distance travelled along the track is computed as follows.

$$objective = D(\bar{m}(t_f)) - D(\bar{m}(t_s)) \tag{2.15}$$

The function $D(s)$ computes the distance along the racetrack (as shown in figure 2.2) (Note that in this case we seek to maximise the objective function). The independent variables are the control values $\bar{u}(t)$ at the time instants $t \in t_s, t_s + t_u, t_s + 2.t_u ... t_f]$ where t_u is the control sampling interval.

The powerful feature of this formulation is that the objective function can be successfully computed for any set of independent variables. Computation of the objective function does not require the attainment of any goal by the vehicle. The displacement and stability constraints are computed from the trajectory resulting from the solution to the IVP, hence the constraint functions are also defined over the entire set of possible independent variable values. This makes for an extremely robust formulation of the problem. The author generally uses this method when working with a particularly unstable vehicle set up (prone to spinning) or when working with a new vehicle model.

2.8.2 Distance Based Formulation

In this formulation, the time taken to travel along the track from a given start point until a particular wayline is crossed is minimised. Computation of this objective also requires the solution to an IVP but in this case it is possible for some independent variable values that the path followed by the vehicle will not cross the wayline. This can happen if the vehicle is steered away from the wayline or if the vehicle enters a 'spin'. Hence for some independent variable values the objective function and some or all of the constraint functions are not defined. The computational procedure handles this problem by introducing a maximum time for completion of the IVP. If this time is exceeded, the routine places a fixed infeasible value into the variables representing any constraints that have not been computed. This causes a very abrupt and non-linear change in those constraint functions. When this problem occurs it is typically during the line search phase (Appendix C) of the NLP code. The NLP code used in this thesis (CFSQP) uses a very simple line search algorithm that is tolerant of this crude approach.

This formulation lacks the ultimate robustness of the time based formulation but compensates to some extent for this with the convenience of being able to specify the locations of the vehicle control discrete values at fixed locations on the racetrack. The control and constraint function points do not then move relative to the racetrack when changes are made to the vehicle set-up.

2.9 Differential Equation Solver

The choice of differential equation solver is driven largely by the requirements of estimating the gradients of the objective and constraint functions using finite differences [50]. A variable step solver may be more computationally efficient but is more likely to introduce numerical noise into the solution. A fixed step size Second order Runge-Kutta solver is used throughout this work. In both time and distance based formulations the equations are solved using time as the independent variable. Since the vehicle will not typically cross a wayline at an integral number of solver timesteps, an interpolation procedure is used to compute the time at which the wayline is crossed. The vehicle models used in later chapters are formulated as ordinary differential equations (ODEs). The method should generalise to models formulated as differential algebraic equations (DAEs) subject to the requirement that the solutions are sufficiently free of numerical noise.

2.10 Choice of Non-Linear-Programming Code

Sequential Quadratic Programming methods [46] have been used effectively in many numerical optimal control problems [32]. The analysis of SQP methods require that the objective and constraint functions have continuous first and second derivatives. Practical implementations of the method vary in their tolerance of such discontinuities. The rate of convergence of SQP methods is strongly influenced by the relative magnitudes of the objective function and constraint derivatives with respect to the independent variables. Test problem sets for NLP codes typically feature problems with objective functions resembling long narrow river valleys. We show later in chapter 3 that even trivial MTM problems have discontinuous derivatives and the tortuous topology of SQP test problems. It is therefore necessary to select an SQP implementation that is very robust. 'C-Feasible Sequential Quadratic Programming' CFSQP [10] is an SQP code that will only evaluate the objective function when the inequality constraints are feasible (less than zero). This allows inequality constraints to be specifed to exclude regions of the objective function that exhibit highly non-linear behaviour. In chapter 4 we show that the MTM problem as formulated here is non-smooth. CFSQP is not stricltly formulated for non-smooth problems but is in practice tolerant of discontinuous derivatives.

2.11 Configuration of CFSQP code

The NLP code CFSQP has a number of configurable set up parameters. There use is discussed below. The reader is again referred to Appendix C for a brief background in NLP.

2.11.0.1 Line Search

CFSQP offers a choice of two line search techniques. The first involves constructing and then searching along a feasible arc. When using this technique CFSQP requires a reduction in the objective function at each major step. Failure to achieve a reduction in the objective will result in termination. In the second approach, the search direction is linear. With this simpler approach, simultaneous feasiblity and descent cannot be guaranteed. In order to

overcome this problem, CFSQP requires that a reduction in objective function must have been achieved in at most four major steps. This non-monotone line search approach is attractive in that it offers the opportunity to go 'uphill' in the optimization process. In circumstances where the optimization is proceeding along the edge of a constrained region, this allows the algorithm to negotiate out of small local minima. For this reason, the non-monotone line search is selected for the MTM algorithm.

2.11.0.2 Repeatability and Precision

There are a number of parameters used to tell CFSQP the level of repeatability that can be expected in the computation of the objective and constraint functions. All of these values are configured to the smallest number representable on the floating point scheme used.

2.12 Modifications and Additions to CFSQP code

In support of the MTM algorithm the author has made a number of modifications and additions to the CFSQP code. A callback interface has been added to allow a user routine to be called at each major step. The callback function is used to monitor and log the performance of the optimizer. Also some additional processing has been added after termination of the optimizer to check the validity of the solution.

2.12.1 Hessian condition computation via SVD

One possible cause of termination of an NLP code is computation of a very small step length due to poor conditioning of the Hessian matrix. We monitor the condition number of the Hessian in the callback function at each major step as follows. The condition number is computed using the singular value decomposition (SVD) [51][24].The SVD is shown in equation 2.16 where G is the Hessian, U and V are orthonormal matrices and w is a diagonal matrix.

$$G = UwV' \tag{2.16}$$

After performing the SVD, the condition number is the ratio of the largest and smallest elements of the diagonal matrix w. This approach to condition number computation is relatively computationally intensive but very robust and accurate. Since in practice, the computation time of the MTM algorithm is dominated by evaluating the objective function and constraints this extra computational load is not significant. When the MTM algorithm is started a long way from an optimal solution, the path to the optimal solution may pass through some extremely non-linear regions. This can result in poor conditioning of the Hessian. In order to combat this, the Hessian matrix is reset to the identity matrix if its condition number exceeds a threshold. This helps CFSQP to find its way through highly non-linear regions.

2.12.2 Verifying Solution Quality

Part of any general purpose NLP code is a procedure to check if the NLP algorithm should terminate. Termination will typically occur for the following reasons:

- The length of the computed search step is too small
- The achieved step length is too small
- The reduction in the objective function during the line search is too small

Termination may occur because an optimal point has been found or because the problem has run into numerical difficulties such as a badly conditioned Hessian or because the objective or constraint functions have discontinuities. In order to establish that the NLP code has terminated at a genuine well defined minimum point the author has augmented the CFSQP code with a procedure to validate the solution. The procedure carries out the following steps:

- Check for suitable smoothness of the objective function and active constraints at the solution.
- Check that the Karush-Kuhn-Tucker (KKT) conditions are satisfied [22].

The first task is carried out for each independent variable $x_n^*, n = 1, 2, ...n_x$ by sampling the objective function and active (and near active) constraints at a number of points around the solution $...x_n^* - 2\Delta, x_n^* - \Delta, x_n^*, x_n^* + \Delta, x_n^* + 2\Delta, ...$ A quadratic polynomial is then fitted through these points and the residual error in the fitted polynomial is logged. This task is designed to check for discontinuities in the objective and constraint functions at the solution. Example plots showing the behaviour of the objective and constraint functions with respect to a single independent variable are provided in the next chapter.

The second task consists of a verification that the Karush-Kuhn-Tucker (KKT) conditions are satisfied at the solution point. This is achieved by computing the matrix A of gradients of the active constraints (by finite differences), computing the pseudo inverse of this matrix A^+ (by SVD [24]) and then multiplying this by the gradient vector of the objective function $\bar{g}$. The result is an estimate of the Lagrange multipliers at the solution point (eqn 2.17).

$$\lambda^* = A^+ \bar{g}^* \tag{2.17}$$

If a is a local minimum then $\lambda_i^* \geq 0, i = 1, 2, ...N$ where N is the number of active constraints at the solution [46].

2.12.3 Cache

The CFSQP interface has separate user functions for computing the objective function and constraints. CFSQP calls the constraint function separately for each constraint. In the case of the MTM problem the objective and all constraint values are evaluated simultaneously by solving one initial value differential equation problem. Typically, CFSQP may call the user objective function followed by several calls to the user constraint functions with the same independent variable values. In order to prevent repeated unnecessary computation we

implement a cache that stores the independent variables, objective function and constraint values each time they are computed. Any subsequent calls made by CFSQP with the same independent variables can then by supplied with objective and constraint values from the cache.

2.13 Finite precision arithmetic issues

In this section we discuss the issues related to the floating point number representation used by the computer running the NLP algorithm. In any finite precision implementation the computed value of the objective function can be considered as in eqn 2.18. Where $n(x)$ represents the 'noise' introduced by the finite precision arithmetic. Let $\Delta\hat{f}$ and Δn be the changes in $\hat{f}(\bar{x})$ and $n(\bar{x})$ resulting from a change Δ in one element x_n of $\bar{x}$. $\Delta\hat{f}$ must be sufficiently greater than Δn in order to discern changes in the objective function for perturbations in the independent variable x_n with magnitudes in the order of Δ.

$$\hat{f}(\bar{x}) = f(\bar{x}) + n(\bar{x}) \tag{2.18}$$

For any given fixed size floating point number format, inaccuracies will result when adding two numbers whose ratio exceeds a threshold. Consider the sum of two numbers $c = a + b$ where $|b| < |a|$ using finite precision floating arithmetic. If the ratio b/a is smaller than the smallest number that can be represented by the mantissa of the floating point scheme then the result of the addition is equivalent to $c = a$. IEEE double precision arithmetic is used throughout this work. This format has a 52 bit mantissa. The largest number representable by the mantissa is 2, the smallest is 2^{-52}. This gives a minimum value on the ratio of approximately 1^{-16}. If the ratio $|b|/|a|$ is of the order of 1^{-16} then only the most significant bit in the mantissa of a will appear in the result c. Hence a will have been effectively quantized to a 1 bit representation for the purposes of the computation. If a is to be represented with any significance then $|b|/|a|$ must be significantly greater than 1^{-16}. For example, if b is to be represented in the sum to IEEE single precision (approximately 5 decimal significant figures) then b/a must be greater than approximately 1^{-10}.

The algorithms employed by SQP non-linear-programming codes typically perform various linear algebra operations such as vector dot product and matrix vector multiplication. Equation 2.19 shows the typical operation of computing the search direction $\bar{s}$ where H is the approximate inverse Hessian matrix and $\bar{g}$ is the vector representing the gradients of the objective function with respect to the independent variables.

$$\bar{s} = -H\bar{g} \tag{2.19}$$

Consider the vector $\bar{v}_n$ representing the elementwise multiplication of the n^{th} row of H and $\bar{g}$. If the ratio of the maximum and minimum values of the non-zero elements of $\bar{v}_n$ exceeds the threshold discussed above. The smallest element will not have any numerical significance in the search direction. This can result from, for example, the ratio of the magnitudes of the largest and smallest elements of $\bar{g}$ being sufficiently large.

2.13.1 Gradient estimation using Finite differences

The NLP code requires estimates of the first derivatives of the objective and constraint functions with respect to each of the independent variables. A simple estimate of the gradient of the function $f(x)$ with respect to x can be computed by finite differences as shown in equation 2.20.

$$\frac{df}{dx} = \frac{f(x+\Delta) - f(x)}{\Delta} \tag{2.20}$$

A suitable choice of Δ is a compromise. We wish to choose a small value for Δ so the second order derivative (curvature) and higher derivatives of $f(x)$ do not have an impact on the gradient estimate. However, as the magnitude of Δ is decreased, the precision of the computed change in the objective function is reduced as explained above (eqn 2.18).

In practice, the impact of this finite difference approximation on NLP performance will depend on the mapping between the independent variables and discretized vehicle controls. Assuming for now a simple one-to-one mapping, a given Δ applied to a control value early in the manoeuvre will produce a larger change in the objective function than a control late in the manoeuvre. Hence a Δ suitable for an early control may be unsuitable for a later control. We consider the selection of a mapping from the discretized controls to the independent variables in the next section.

In practice, it is quite easy to determine if errors in gradient computation are having an impact on the NLP outcome. It is simply a case of solving the same problem with several different choices of finite difference step(s). If the solutions or the path to the solutions have significant differences then the gradient estimates can be suspected.

Casanova [8] uses automatic differentiation for the computation of the derivatives. This is a computational procedure whereby the source code of the model is analysed by a piece of software that then analytically computes gradients by repeated application of the chain rule. This approach is not used here as the author is motivated to develop an MTM method that is capable of operating with the model as an unseen 'black box'.

2.13.2 Basis and Scaling for the independent variables

The discussion so far has assumed a simple one-to-one mapping between the discretized vehicle controls and the independent variables of the NLP problem. We consider here the process of producing the independent variables by applying a linear transform to the discretized vehicle controls. The inverse transform is applied to the independent variables to produce the vehicle controls for computation of the objective and constraint functions. This process can be considered as one of finding the basis in which the independent variables represent the vehicle controls. As previously stated, a similar change in control values early in the manoeuvre will have a larger effect on the manoeuvre time than that later in the manoeuvre. Hence the gradient of the objective function will be significantly different with respect to controls early and late in the manoeuvre. The same is true for constraint functions. This can lead to numerical problems as previously discussed. This problem will worsen as the manoeuvre length is increased. In the case of the search direction computation (eqn 2.19), the problem can be caused by the gradient vector $\bar{g}$ or the inverse Hessian approximation H. We can to some extent reduce problems in $\bar{g}$ by applying appropriate

scaling factors to the discretized controls. In practice, the maximum manoeuvre length that can be solved by the MTM procedure is not limited by these factors. The main limiting factor is the increase in highly non-linear behaviour of the objective and constraint functions as the manoeuvre size increases. Examples of the non-linear and discontinuous objective and constraint functions are given in section 3.3.3. The basis used in most of the results in this thesis consists of a simple one-to-one mapping between independent variables to steer angle in radians and throttle/brake in normalised range +/- 1.

2.13.3 Fourier Basis

As previously discussed the early controls have a larger impact on the objective function and constraints. In order to investigate the impact of this phenomenon on the algorithm we optionally apply a change of basis to the driver controls using a linear transformation. A discrete Fourier transform is applied to the independent variables before the objective function is computed. The driver controls consist of two arrays, the N_s element steer array u_s and the N_T element throttle/brake array N_T.

$$u_s(n)...n = 0, 1...N_s - 1 \tag{2.21}$$

$$u_T(n)...n = 0, 1...N_T - 1 \tag{2.22}$$

The transformed throttle controls are computed as follows (steer is similar):

$$u_T^*(k) = \Sigma_{n=1}^{N_T} \cos(\frac{2\pi kn}{N_T})...k \in [0, 2, 4, 6...] \tag{2.23}$$

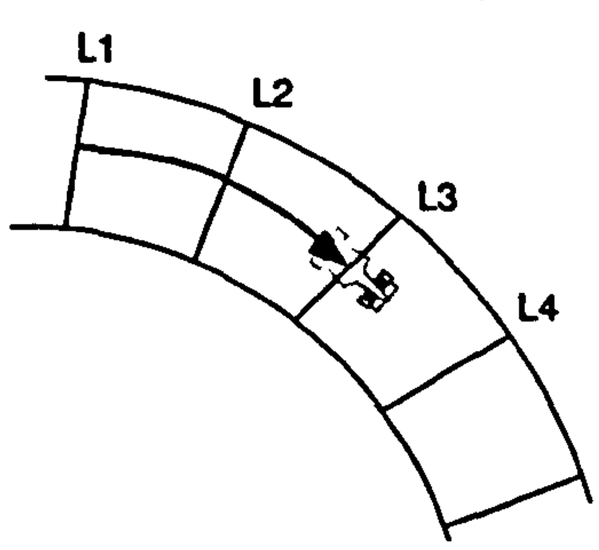
$$u_T^*(k) = \Sigma_{n=1}^{N_T} \sin(\frac{2\pi kn}{N_T})...k \in [1, 3, 5, 7...] \tag{2.24}$$

When this transformation is applied, the optimizer sees the Fourier coefficients of the controls rather than the controls themselves. Each independent variable now has an impact on all of its corresponding discretized control inputs. Hence each independent variable contributes to all of the constraint values. Results obtained with this method are described in section 3.3.1. When the Fourier basis is used, an additional set of constraints is used to bound the time domain representation of the driver controls.

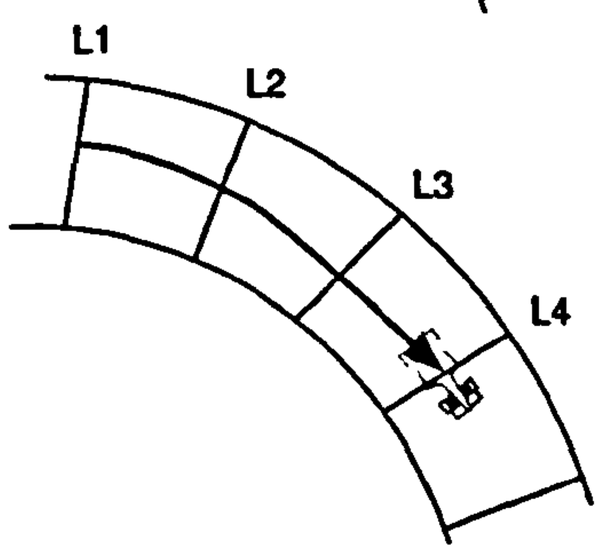
2.14 Generation of Initial Feasible Solutions

In order to build confidence that a globally optimal solution has been found, it is necessary to search as much of the vector space occupied by the independent variables as possible. In order to achieve this aim, we initialise the independent variables with values from a uniform random number generation routine [51]. Randomly generated initial values will not typically satisfy the displacement or stability constraints. The first task of the algorithm is to move from the initial control values to a feasible set before optimization can begin. CFSQP can be configured to find a feasible point without regard for an objective function. In early development of the algorithm this process was attempted in a single step. That is, all of the independent variables were initialised then a single call to CFSQP was used to find a feasible point. This problem with this approach is that when stability constraints are violated, the

First find feasible point between L1 and L2...



Next find feasible point between L1 and L3...



Next find feasible point between L1 and L4...

Figure 2.5: Incremental procedure for finding feasible points

constraint functions can become highly non-linear. In addition, an NLP code will typically attempt to go 'downhill'. When in search of feasiblity it will be searching for a new point that is more feasible than the current point. This will cause problems if from where we are now it is necessary to become less feasible on the path to becoming feasible. Numerical experiments indicated that local minima exist in the infeasible area. This makes it very difficult for CFSQP to make progress toward a feasible point. To counter this problem, an incremental procedure was developed that allows a feasible point to be found in stages. The description below relates to the distance based method but generalises to the time case.

In general, the controls at the first wayline in a manoeuvre are set to fixed value and do not form part of the optimization. We then move to the next wayline that has an active control and/or steer value. We then find a feasible point for the short manoeuvre from the first wayline to the second. We then move on to the next active point and find a feasible point for a manoeuvre between the first wayline and the third. This process is repeated until the final wayline is encountered. This proves to be a very robust method even when working with very unstable vehicles. The process is summarised in figure 2.5. The resulting feasible point is typically very far from optimal and this allows a wide range

of the vector space occupied by the independent variables to be explored on the way to an optimal solution.

2.15 Computational Procedures

The MTM method described here forms the core algorithm of a larger piece of software developed by the author under the name **LORA** (**L**aptime **O**ptimisation for **R**acecar **A**nalysis). A brief description of the design and implementation of this software is given in Appendix F. The software supports the concept of an 'experiment'. An experiment typically consists of repeatedly computing a minimum time solution to a given problem. The process that generates each solution is termed a 'run'. Each run typically begins by seeding the initial values for the driver controls from a random number generation routine. Hence an experiment typically consists of solving an MTM problem from a number of different start points. This process is used to determine the basic accuracy and repeatability of the MTM method and to gain confidence that the MTM solutions represent global minima. An experiment is specified by a number of parameters:

1. manoeuvre shape and length
2. Vehicle model and set up
3. Tyre model and set up
4. Range of initial values for the controls and the parameters for the random number generation scheme.
5. Time or distance method to be used
6. Discretization interval and location for controls
7. Discretization interval and location for constraints
8. Vehicle state constraint limit values
9. Number of runs to perform

In addition, each run consists of one of the computational procedures outlined below. An example procedure is shown in figure 2.6. The choice of computational procedure is determined by the trade-off required between the elapsed time consumed by the procedure and the desire to comprehensively search the solution space to achieve high confidence that the solution is globally optimal. Procedure A is used for all of the final results presented in this thesis.

2.15.1 Procedure A

Procedure A is designed to search as much as possible of the space occupied by the vehicle control vector. Its characteristics are:

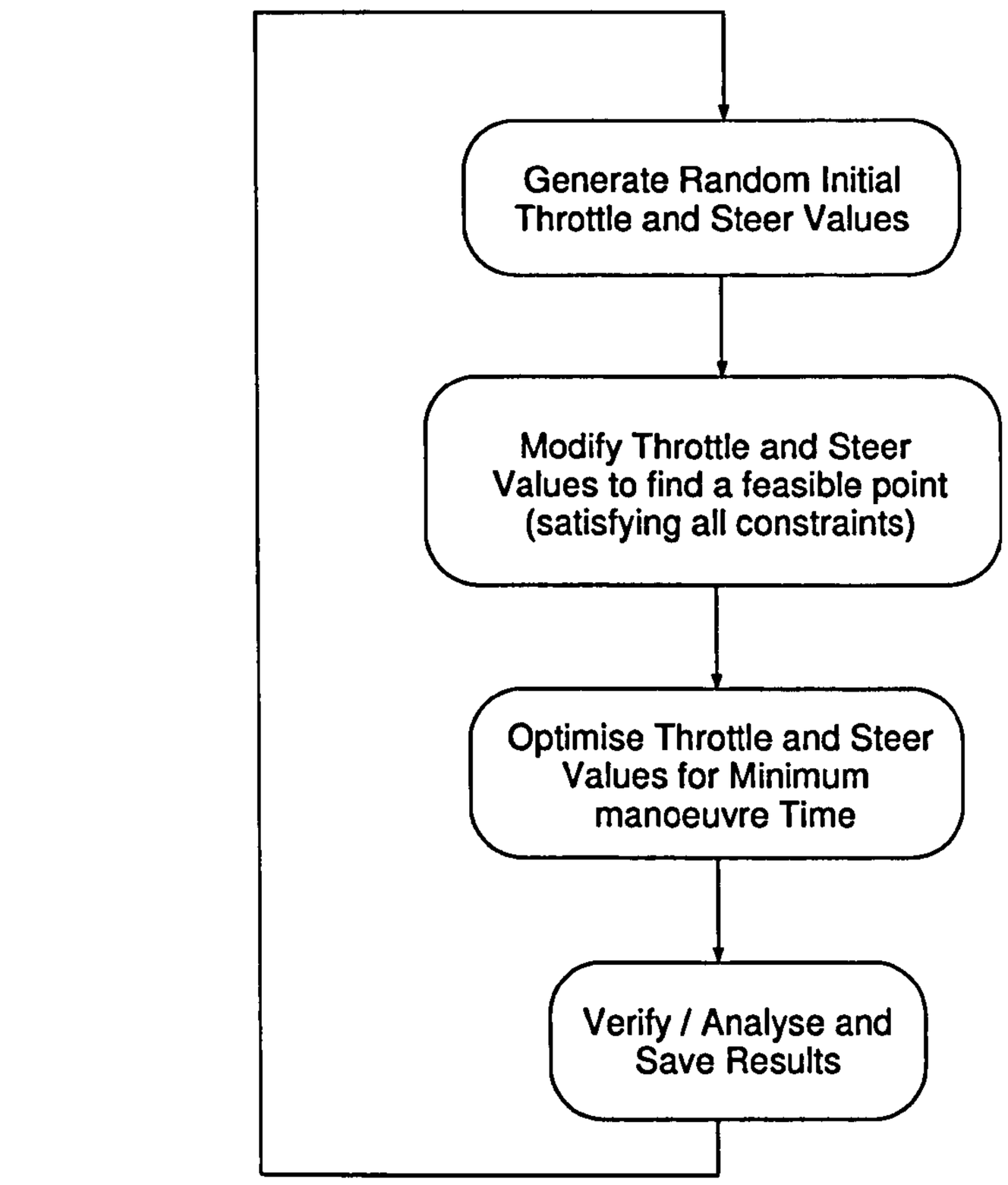


Figure 2.6: Example Computational Procedure

- Bounds on vehicle controls are set to their maximum values. For throttle/brake this represents the values +/- 1. For steer this is a range that guarantees that bounds will not be active at the solution, typically +/- 15°.

- Initial random control values are taken from a uniform distribution with bounds equal to the full control bounds. That is, an initial control value can assume any value that satisfies the bounds for that control.

The procedure by which the initial random control values are modified to satisfy all contraints (find a feasible point) is described in section 2.14. All of the results in this chapter are generated using procedure A.

2.15.2 Procedure B

If a comprehensive search of the solution space is not required, a solution to the problem can be found using less computer time as follows. A previously computed 'reference' solution to the problem (possibly using procedure A) is used to generate a set of bounds on the vehicle

control values. The bounds are set in a window around the reference solution. Suitable freedom can be allowed to increase the bounds on the manoeuvre at sensitive points such as initial braking. Initial values for the controls are then selected from a uniform distribution with bounds equal to the control bounds.

2.15.3 Procedure C

When working with closely spaced control points (higher bandwidth control systems) the time taken to solve the MTM problem increases. For the purpose of some experiments we are more interested in the time to reach a solution rather than an extensive search of the solution space. In this case, a bandwidth recursive approach is used. For example, when finding a feasible point for a 4m control spacing problem, a feasible point can first be found for an 8m problem. The initial values for the 4m spaced problem can then be found by interpolation of the final values from the 8m spaced problem.

Chapter 3

Performance of the MTM Algorithm

> Coming off an Apex is relatively easy, it's merely like balancing on a tighrope...but going into the Apex is a little worse...it's like leaping onto the tightrope blindfold
>
> Paul Van Valkenburgh

3.1 Summary

In this chapter we apply the method described in chapter 2 to a number of test problems using the quasi-steady-state vehicle model described in Appendix A and a simple manoeuvre consisting of a single 90° corner. We provide a detailed analysis of the optimized solutions to this manoeuvre. We demonstrate the accuracy and convergence behaviour of the method. We also show that the MTM problem, as formulated here, possesses multiple locally optimal solutions. The optimal nature of the solution is validated through the computation of Vehicle stability derivatives (Appendix E) and tyre efficiency diagrams (Appendix B). We go on to consider the influence of the control discretization interval (time or distance) on the manoeuvre time.

As should be expected, the method is able to exploit the extremes of vehicle performance in a manner that could not be achieved by a human driver. The human driver operating in the real world needs to leave some stability margin in the vehicle to account for uncertainties in the environment. By making some assumptions about the performance limits of the human driver, we attempt to emulate driver behaviour by adding additional constraints to the method. We constrain the minimum vehicle yaw stiffness derivative and steering control moment derivative achieved during the manoeuvre (Appendix A). We also consider a suitable control discretization interval that is representative of the human driver. It is beyond the scope of this work to study human factors in any detail or to correlate the MTM with human driver behaviour.

3.2 Manoeuvre, Vehicle and tyre model configuration

The vehicle model used is described in Appendix A. The model has a number of parameters specifiying the mechanical and aerodynamic set-up of the vehicle. A particular set of parameters wil be referred to as a 'set-up'. A number of named reference set-ups are given in Appendix A. The tyre model also has a number of parameters and a number of reference tyre configurations is provided in B.

3.2.1 Summary of tyre and vehicle 'set-ups'

Two tyre types are described in Appendix B. Tyre set 1 has similar levels of longitudinal and lateral performance at the front and rear. The rear peak lateral force is achieved at a slightly smaller slip angle than the front. Tyre set 2 has less rear lateral grip than front. The rear peak lateral force is achieved at a significantly smaller slip angle than the front. Tyre set 2 also has a more rapid drop in grip after the peak in the μ / slip curve. Running tyre set B generally results in a less stable vehicle.

QSS car set-up A has a relatively rearward roll moment distribution (RMD). QSS car set-up B has a relatively forward RMD with set-up C somewhere in the middle. Tyre set A tends to produce an understeer balance and this is reduced by set-up A. Running Car set-up B with tyre set 2 results in a car that is prone to snap oversteer on corner exit. Various combinations of tyre and vehicle set-up are used in this and the following chapters to show the impact of vehicle/tyre characteristics on the MTM method.

3.3 A detailed analysis of a single 90° corner

In this section we present some results for the righthand corner manoeuvre shown in figure 3.1. The MTM algorithm and computational procedure A was used to generate 100 solutions to the problem. The problem configuration is described by table 3.2. The vehicle state limits are chosen so that (with the exception of minimum wheel load) the state constraints are not active at the solution. Figure 3.1 shows the optimized vehicle path.

Item	Configuration 1	Configuration 2	Configuration 3
Car	QSS	QSS	QSS
set-up	QSS/A	QSS/B	QSS/C
Tyres	ST/A	ST/B	ST/B
Differential	Locked	Locked	Locked

Table 3.1: Vehicle/Tyre Configurations

3.3.1 Solution repeatability and existence of local minima

The resulting manoeuvre times consist of two distinct solutions. The manoeuvre times for five of each solution are shown in Table 3.5. Each solution has been tested for optimality (as explained in 2.12.2). This demonstrates that at least two solutions to the problem exist.

Parameter	Value
manoeuvre Length	250m
manoeuvre	Righthander
MTM Method	Distance
Control Spacing	Steer: 5m Throttle: 5m
Trackwidth	10m
Initial Vehicle CG XY	0,4.9
Initial Speed	60 m/s
Minimum Wheel Load	1N
Front Slip Ratio Limit	+/- 50 %
Rear Slip Ratio Limit	+/- 50 %

Table 3.2: Test Configuration RH/1

Parameter	Value
Control Spacing	Steer: 10m Throttle: 10m (Steer value held at zero until 55m)

Table 3.3: Test Configuration RH/2 (Only differences to RH/1 shown)

Parameter	Value
Control Spacing	Steer: 5m Throttle: 5m (Steer value held at zero until 55m)

Table 3.4: Test Configuration RH/3 (Only differences to RH/1 shown)

Solution 1 Manoeuvre Time	Solution 1 Final Speed	Solution 2 Manoeuvre Time	Solution 2 Final Speed
5.09557890	60.36551846	5.10524612	60.08085658
5.09558203	60.36535574	5.10524612	60.08098167
5.09557872	60.36567244	5.10524616	60.08195057
5.09557873	60.36541096	5.10524617	60.07956579
5.09559235	60.36551804	5.10524612	60.08082346

Table 3.5: Two Solutions to 250m single corner

Of the 100 optimizations carried out, 64 consisted of the faster solution. The experiemt was repeated using a Fourier basis for the controls (Section 2.13.3). The same two solutions were found using this basis. Table 3.6 shows the manoeuvre times for five of each solutions in this case.

It can be seen that in the case of this vehicle, tyre and manoeuvre the optimized solution times have a spread of < 1 ms or approximately 0.02%. During the course of this chapter and the next, we present many results using different manoeuvres, vehicle set-ups and tyres. In order to allow a thorough investigation of solution repeatability, these solutions

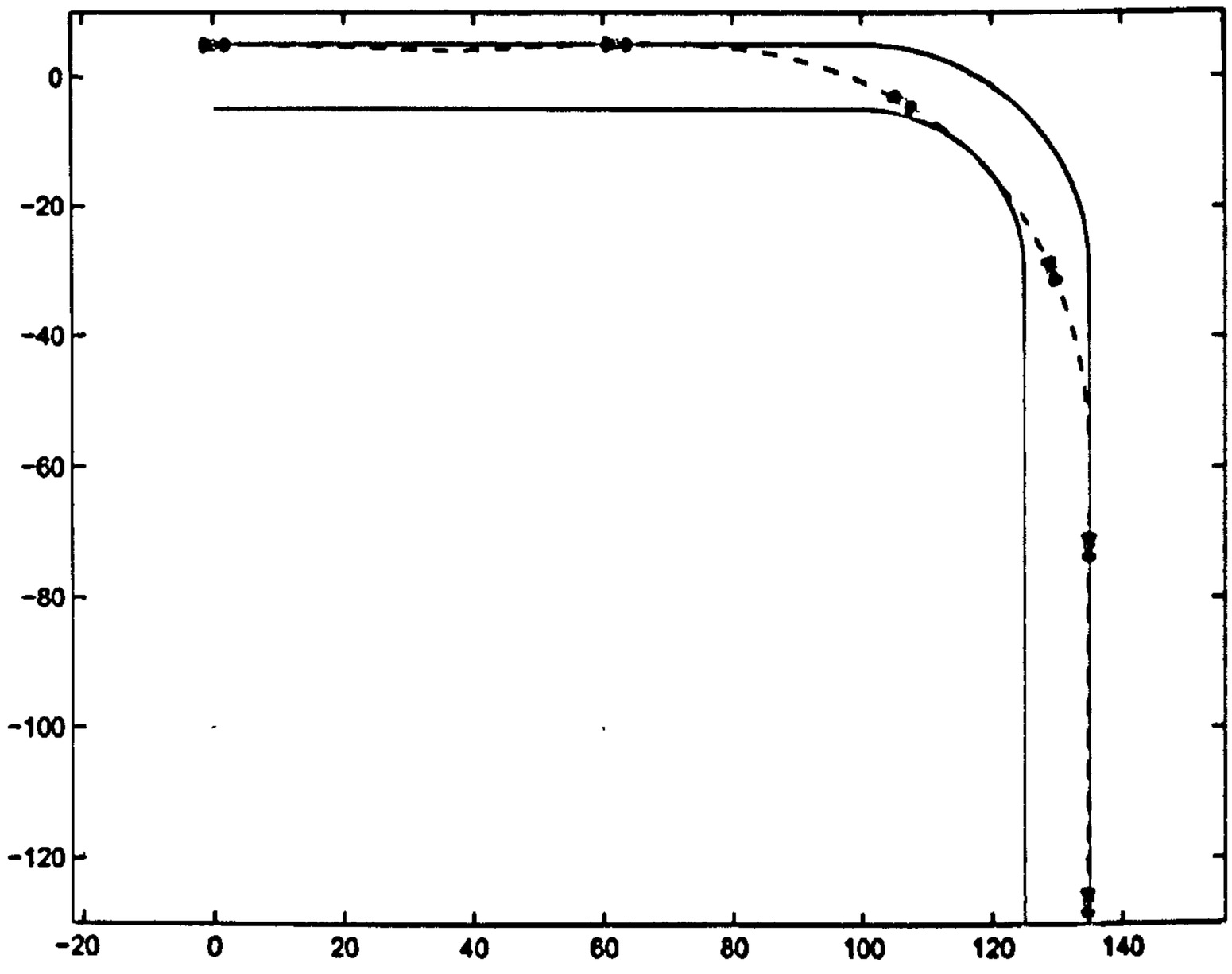


Figure 3.1: Driven Path

Solution 1 Manoeuvre Time	Solution 1 Final Speed	Solution 2 Manoeuvre Time	Solution 2 Final Speed
5.09557548	60.36548978	5.10524428	60.08036605
5.09557577	60.36491595	5.10524288	60.08070747
5.09557531	60.36559319	5.10524289	60.08205141
5.09557549	60.36588355	5.10524276	60.08206115
5.09557555	60.36518939	5.10524292	60.08060181

Table 3.6: Two Solutions to 250m single corner with Fourier basis for controls

are generated using computational procedure A. Hence many solutions are generated to each problem from widely differing initial points. Results are then typically presented showing the repeatability and any local minima found.

3.3.1.1 Comparison of Locally optimal solutions

The throttle and steer values for both solutions are shown in figure 3.2. The location of the control sample points is shown along the bottom of the plots. The speed and timeline comparison are shown in figure 3.3. The timeline indicates the time difference between the two solutions with distance travelled along the driven path as the independent variable. The timeline shows that the slower solution arrives at the apex of the corner in less time and is approximately 30ms faster at some stage just beyond the minimum speed point. The faster solution then gains back time as it is able to apply driving torque earlier in the manoeuvre.

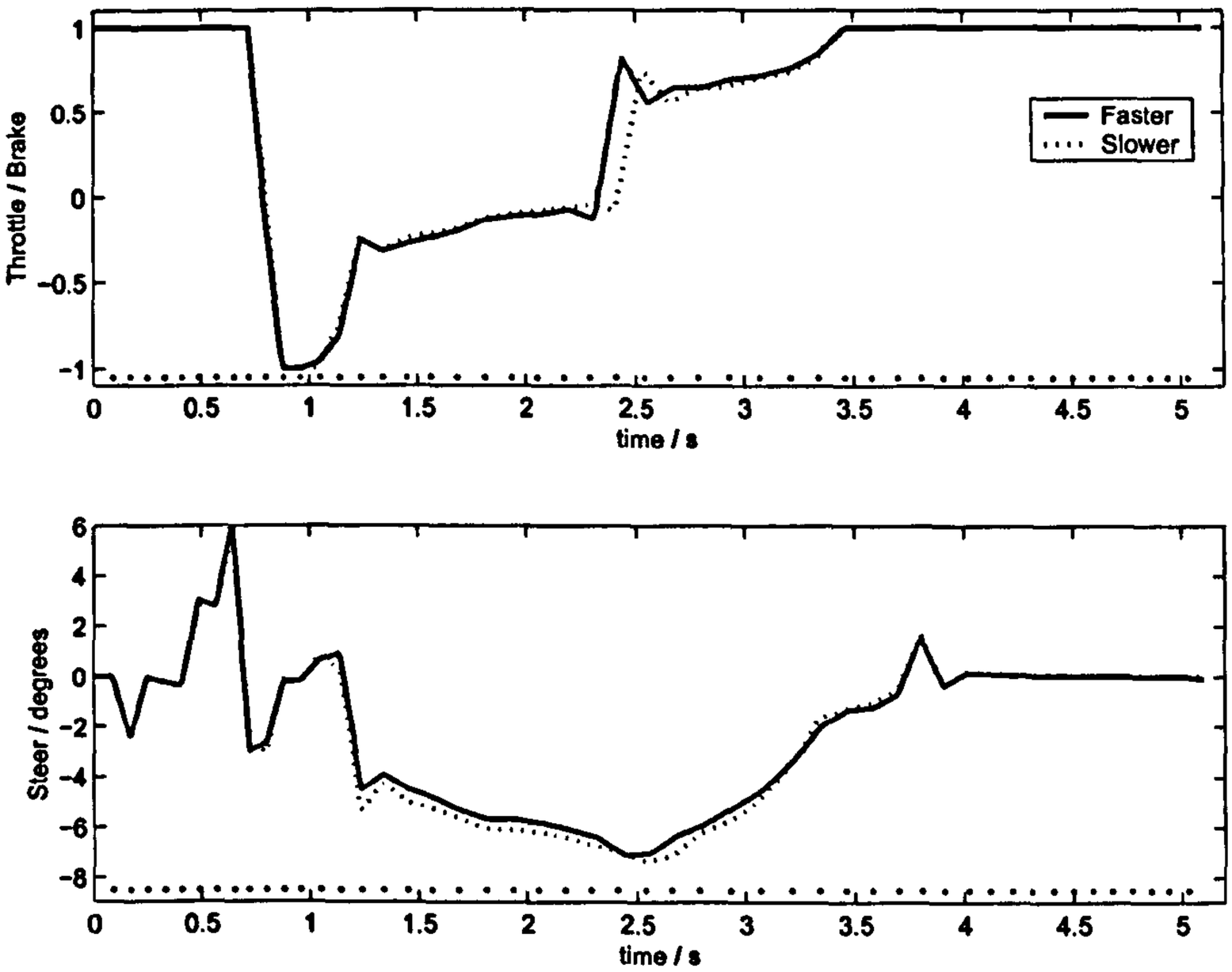


Figure 3.2: Throttle and Steer Comparison

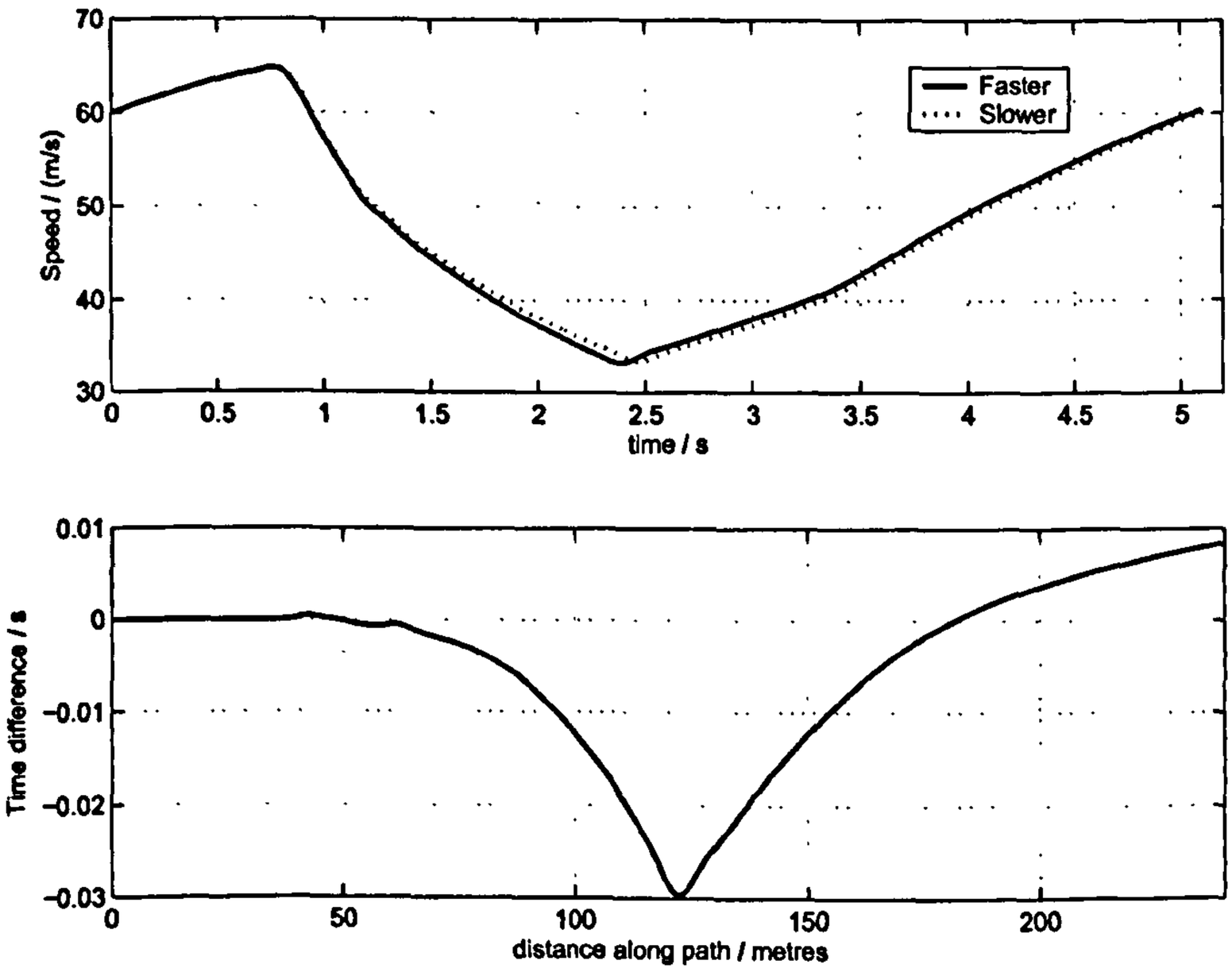


Figure 3.3: Speed and Timeline Comparison

The faster solution also has the higher final speed, an important consideration if this corner is followed by a straight section of track.

3.3.1.2 Resolution

The optimal manoeuvre times computed can be considered as a 'true' optimal time plus a numerical error term due to finite precision implementation and NLP code termination criteria (see section 2.13).

$$t_{computed} = t_{optimal} + t_{noise} \tag{3.1}$$

The most simple and effective method for determining the resolution of a lap time simulation code is to plot the optimal manoeuvre time against vehicle mass (we consider the effect of vehicle mass and other vehicle parameters in more detail later). An increase in vehicle mass should always result in an increase in manoeuvre time. Consider a small change of mass Δm with a corresponding change in manoeuvre time δt. As Δm is reduced there should come a point where t_{noise} exceeds δt. In figure 3.4 we consider steps in vehicle mass of 1 gramme. The plot clearly shows that even for this short manoeuvre, the method is capable of resolving to less than one gramme of vehicle mass.

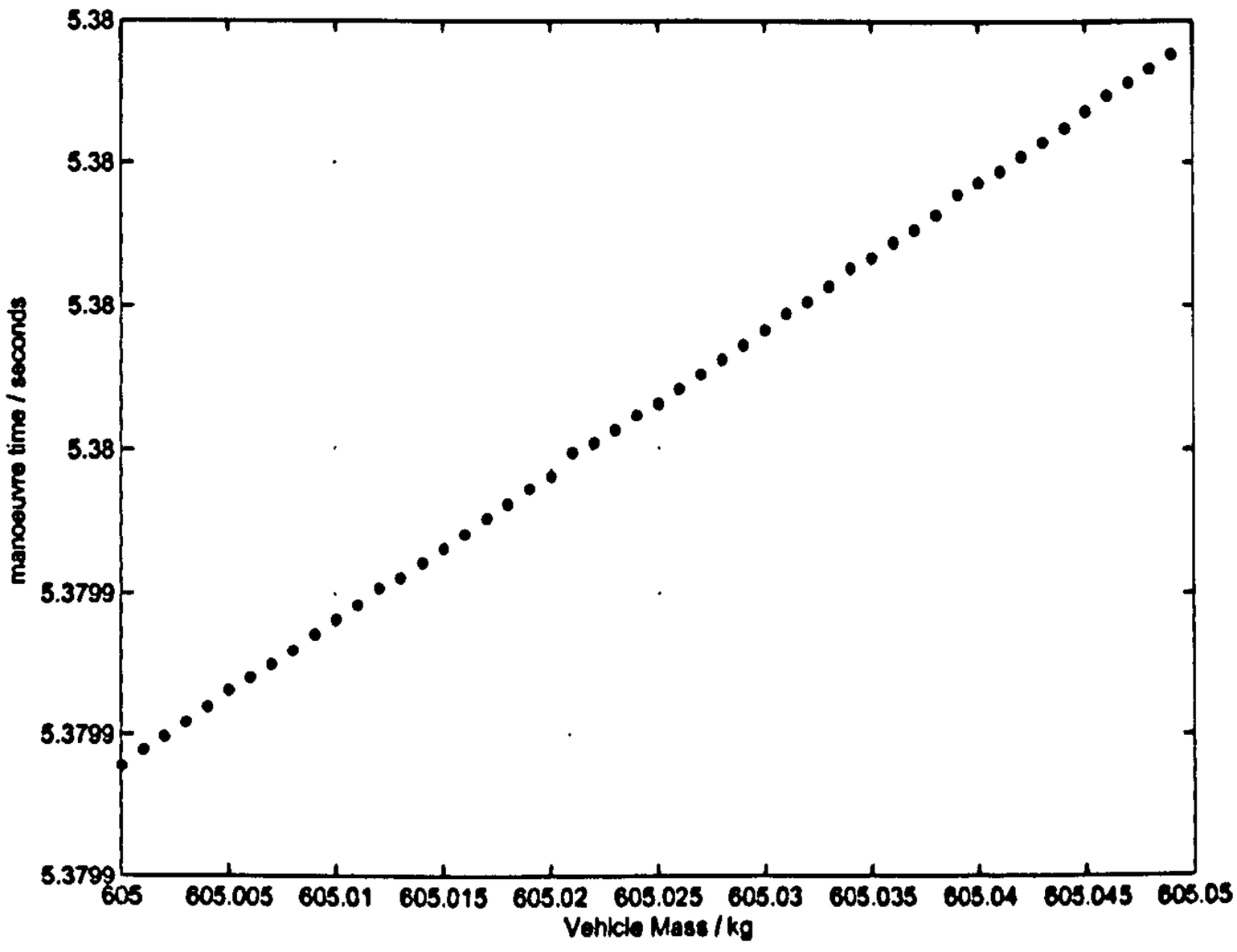


Figure 3.4: Effect of 1 gramme steps in Vehicle Mass on manoeuvre time (Vehicle QSS/B, Tyre TS/B, Config RH/1)

3.3.1.3 Influence of Vehicle Stability on Solution Spread

In the above examples, the method is shown to produce very accurate results. Here we consider the effect of destabilising the vehicle on the spread of optimal solution times. This is achieved by applying a scaling factor to the rear tyre lateral and longitudinal forces (Appendix B). A reduction in rear grip is associated with exponentially divergent unstable vehicle behaviour. Figure 3.5 shows the resulting optimal manoeuvre times at grip scale levels from 1.01 to 0.95 for two vehicle and tyre configurations. The results are obtained

with the experimental set-up RH/2. In figure 3.5 (a) it can be seen that in the grip scale range greater that 0.99 the solution spread is small (the time differences between multiple locally optimal solutions in this region is too small to see on this scale). At the 0.99 point two distinct solutions can be seen. Between 0.99 and 0.98 two solution families exist. For values less than 0.98 the number of solutions starts to increase. Figure 3.5 (b) shows that as the rear grip level reaches 0.975 the solution spread starts to increase. This data demonstrates that as vehicle stability is reduced solution spread increases. The same effect can be introduced by other vehicle parameter changes that tend to promote instability such as rearwards centre of gravity location or forwards aerodynamic balance. The effect of a reduction in rear grip is analysed in more detail in the following chapter.

3.3.2 Rate of Convergence of Objective Function

An example of the improvement in manoeuvre time against the optimizer major step number is shown in figure 3.6. The manoeuvre time is within 0.1% (5 ms) of the optimal value after 200 major steps. The final 5 ms of objective improvement requires over 180% of the computational resources. Figure 3.7 shows a detail of figure 3.6. The plot shows that the rate of convergence is subject to rapid change and has 'shelves' where progress is very slow. This fact must be taken into account when designing a termination algorithm for the optimizer (Appendix D). Any such algorithm that operates on the basis of objective progress may result in premature termination. CFSQP terminates when the computed step length (or step taken achieved) is less than a user programmable value. For this work, this value is set to the smallest number representable in the floating point scheme used.

Figures 3.9 and 3.8 show the history of a steering and throttle/brake control value against major iteration number from 100 different start points. These plots give an indication of the range of control values that are explored en route to the optimized solution. The throttle plot is taken from an experiment having two locally optimal solutions. In each of these experiments, the range of the pseudo random number generation routines is set to +/- 1 for the throttle/brake and +/- 25° for the steer. The scheme used to generate initial feasible points (section 2.14 is responsible for the reduction in these ranges to those seen in the figures 3.9 and 3.8.

3.3.3 The nature of the objective and constraint functions

As described in chapter 2 and appendix C, NLP codes are designed to operate with smooth non-linear functions. In this section we provide a limited analysis of the smoothness of the objective and constraint functions used by the MTM method. The objective and constraint functions cannot be easily visualised due to the large number of independent variables. Here we study the sensitivty of these functions to a change in a single independent variable around its optimal value. Figures 3.10, 3.11, 3.12 show this for a number of throttle/brake and steer independent variables. The location of the optimal value is highlighted. Figure 3.14 shows a similar plot for example minimum wheel load and displacement constraints. It is evident from this data that both the objective and constraint functions exhibit highly non-linear behaviour.

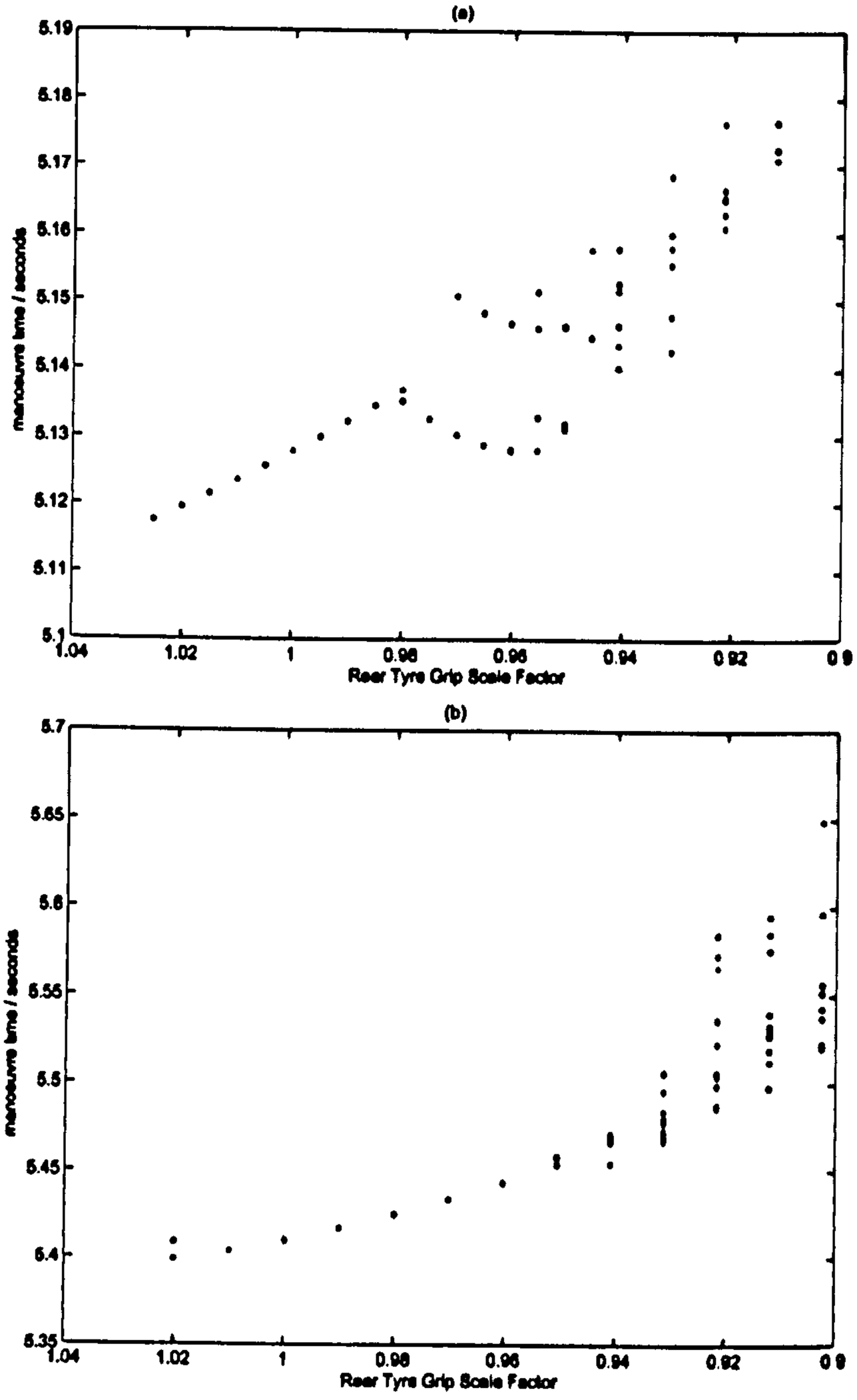


Figure 3.5: Influence of Rear Grip on Solution Spread (a) QSS/A,TS/A (b) QSS/B,TS/B

3.3.3.1 An additional simple example showing the nature of the objective function

In this section we analyse a very simple MTM problem having only two independent variables. The manoeuvre is simple straight line acceleration over a distance of 10 metres with an initial vehicle speed of 20 m/s. This speed is sufficiently low that the vehicle performance is tyre limited. The manoeuvre consists of three parallel waylines spaced by 5 metres. The throttle/brake value at the first wayline is set to 1. The steering values at each wayline are held at zero. The two independent variables represent the throttle values at the second and third waylines. Figure 3.15 shows contour and surface plots of the objective function

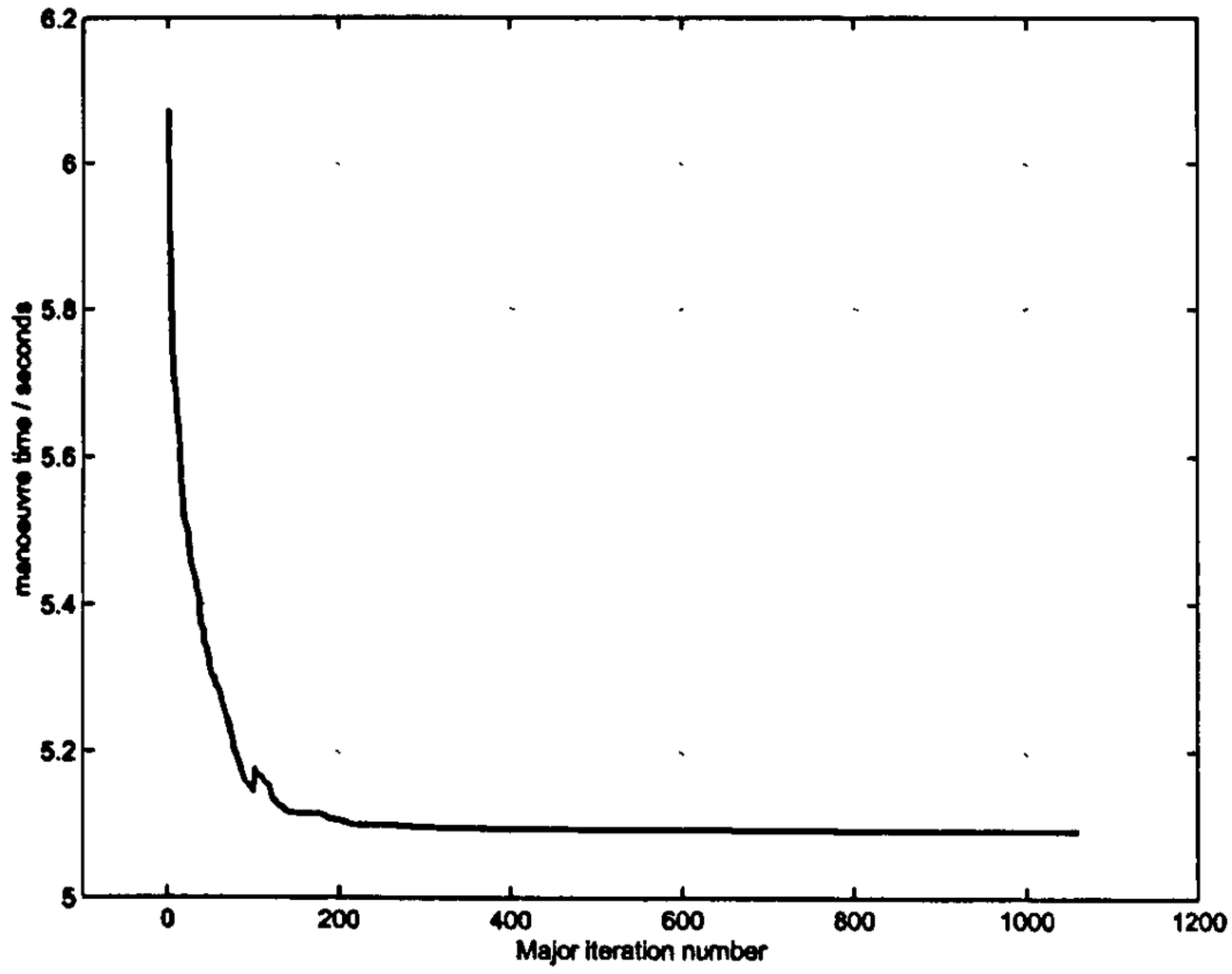


Figure 3.6: Objective Progress

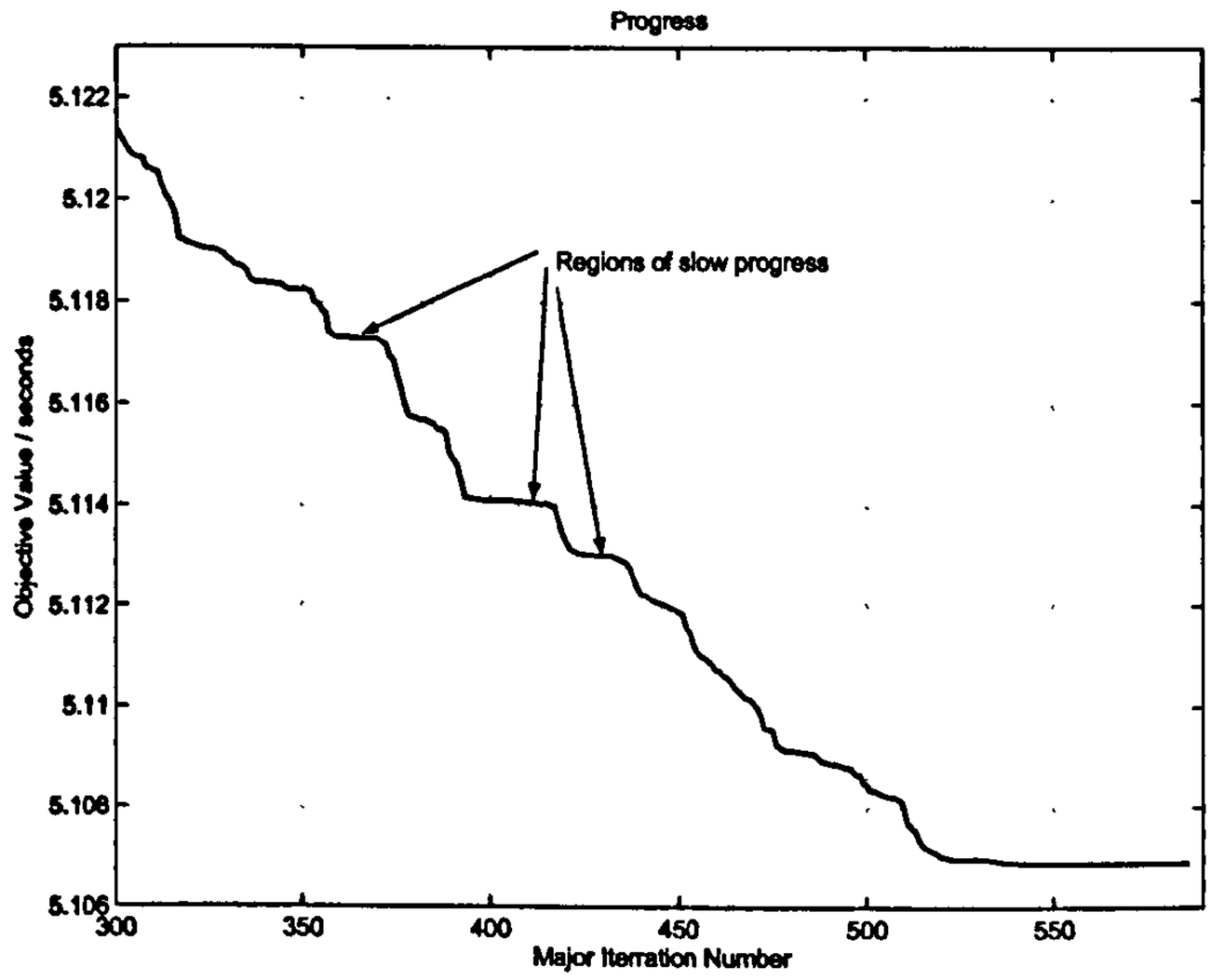


Figure 3.7: Detail of Objective Progress

(time to travel 10m) against the independent variables. Figure 3.15 also shows a cross-section of the objective function parallel to each of the independent variable axis through the minimum point. This data clearly shows that despite the simplicity of the manoeuvre, the objective function is both non-smooth and has a significant spread in gradients along the two axes. The optimal solution lies in a point along a 'crease' in the objective function.

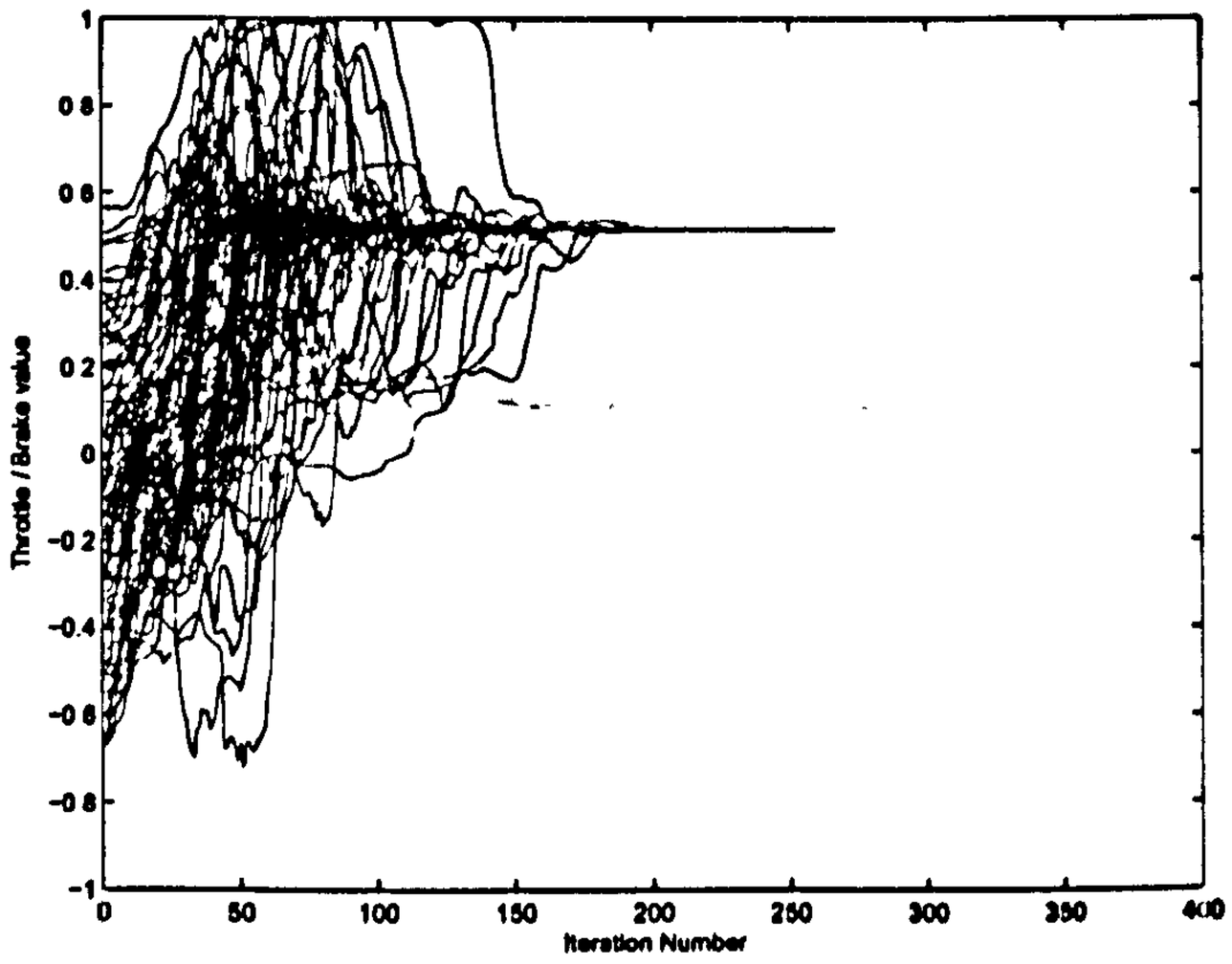


Figure 3.8: Example of Throttle Value Convergence in a problem with a two solutions

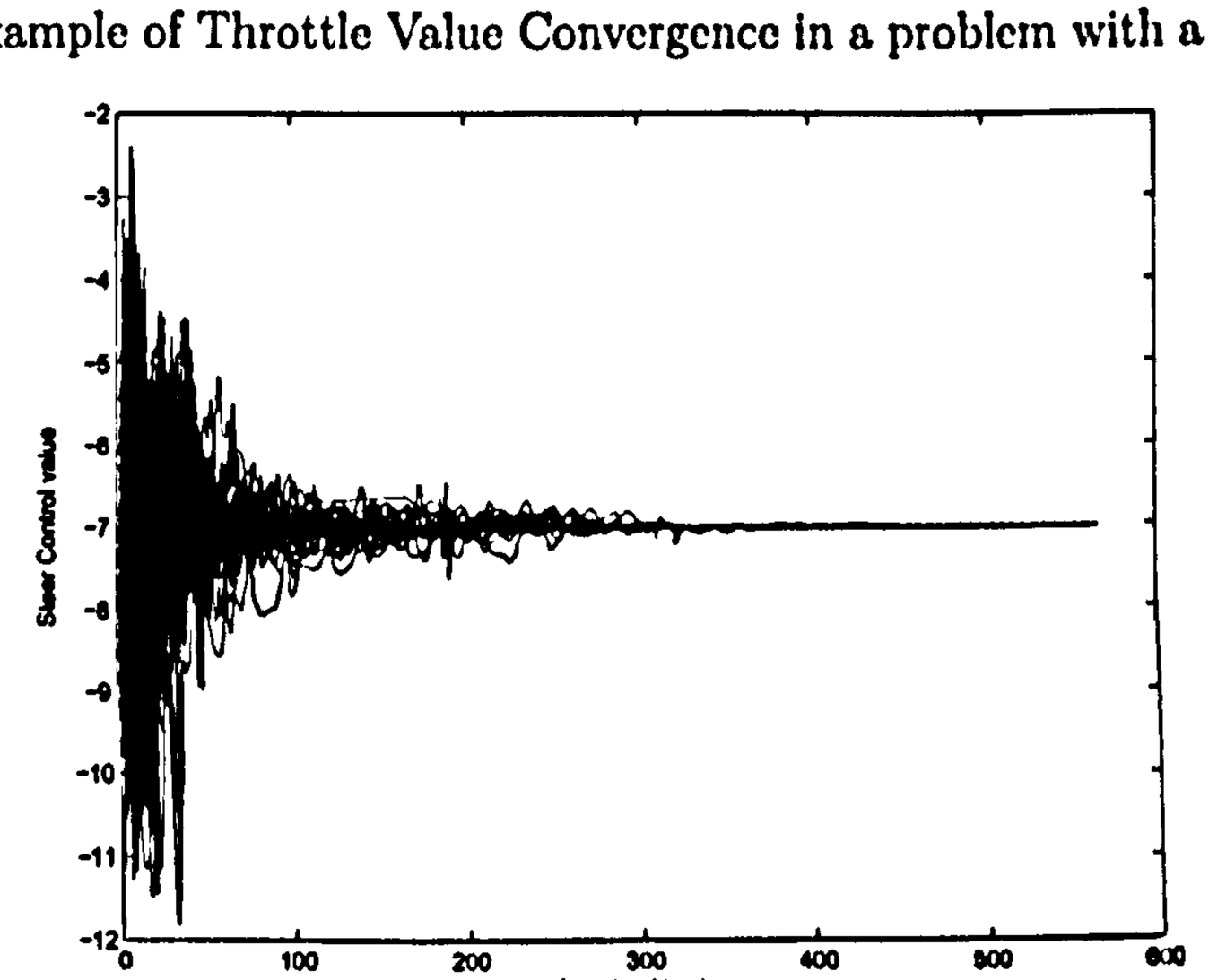


Figure 3.9: Example of Steer Control Value Convergence in a problem with a single solution

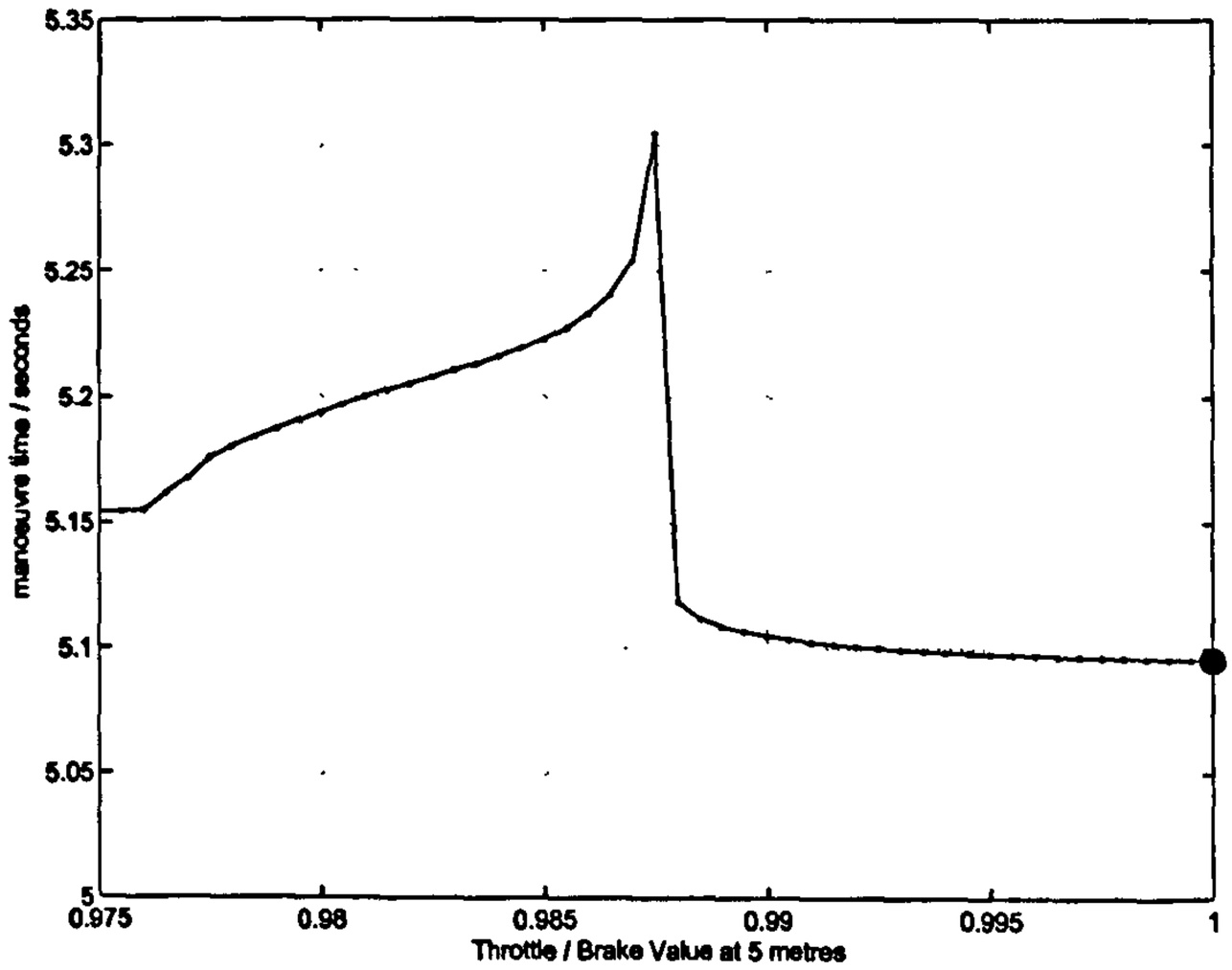


Figure 3.10: Effect of Throttle value at 5m on Objective Function

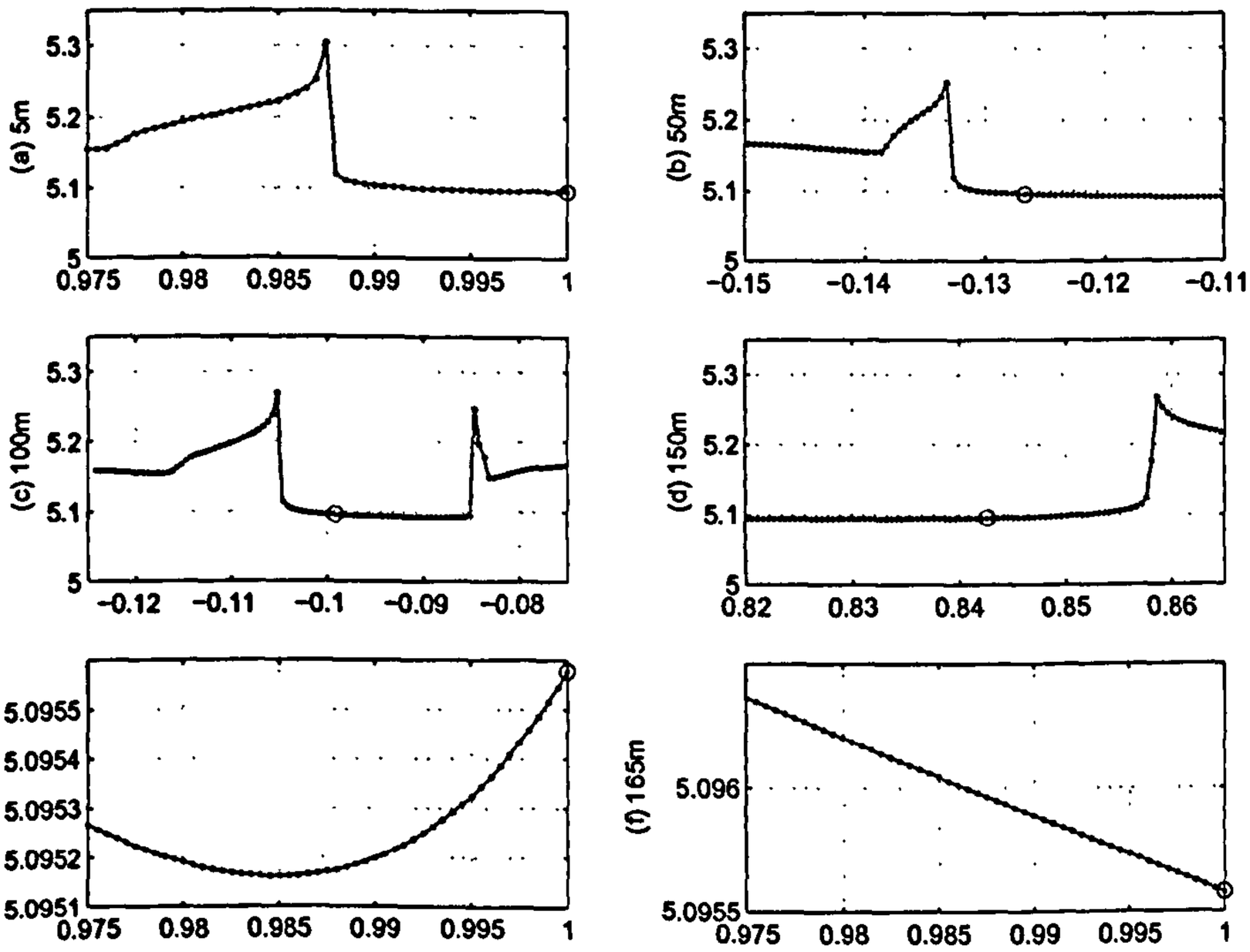


Figure 3.11: Effect of individual Throttle control point values on Objective Function

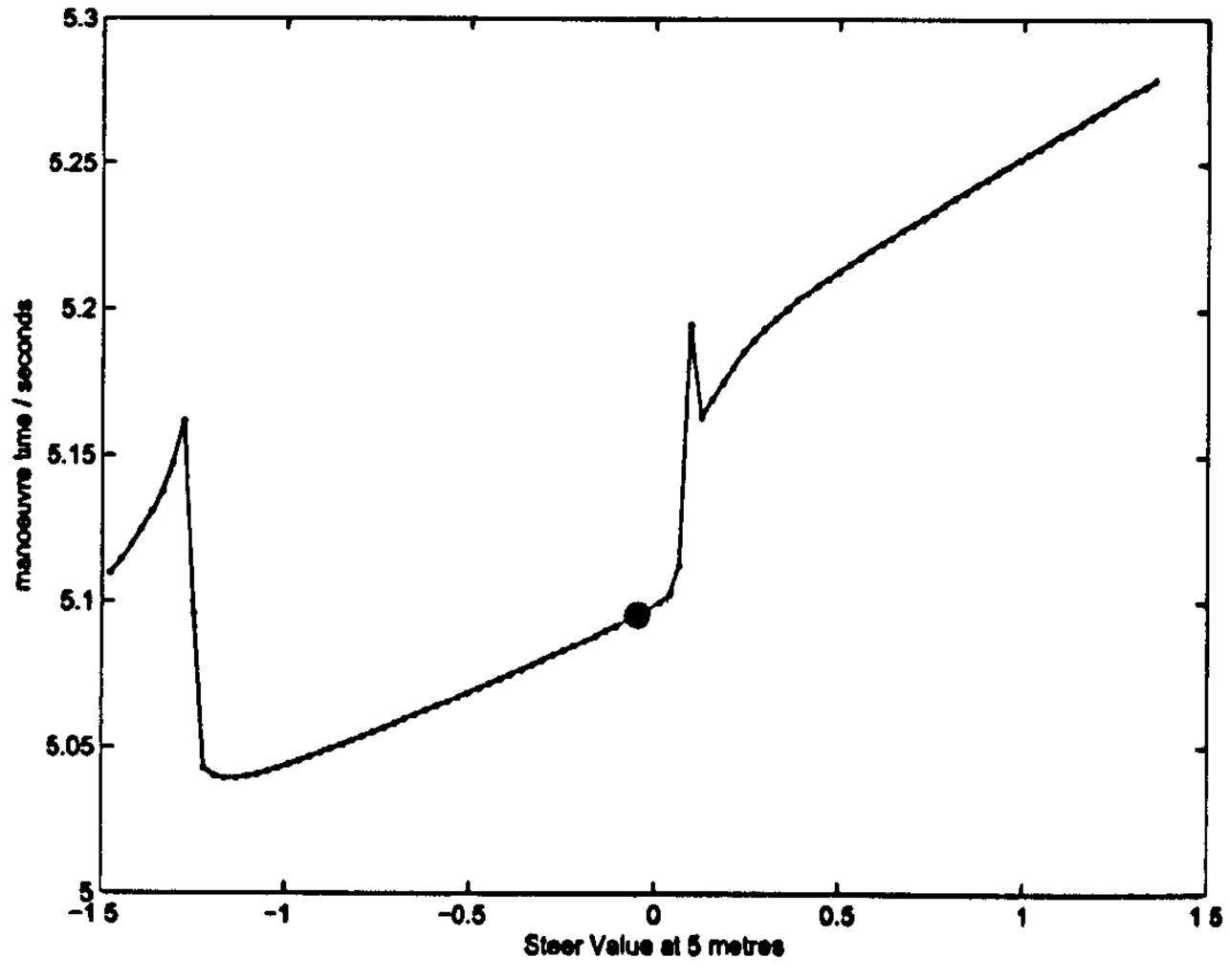


Figure 3.12: Effect of Steer value at 5 metres on Objective Function

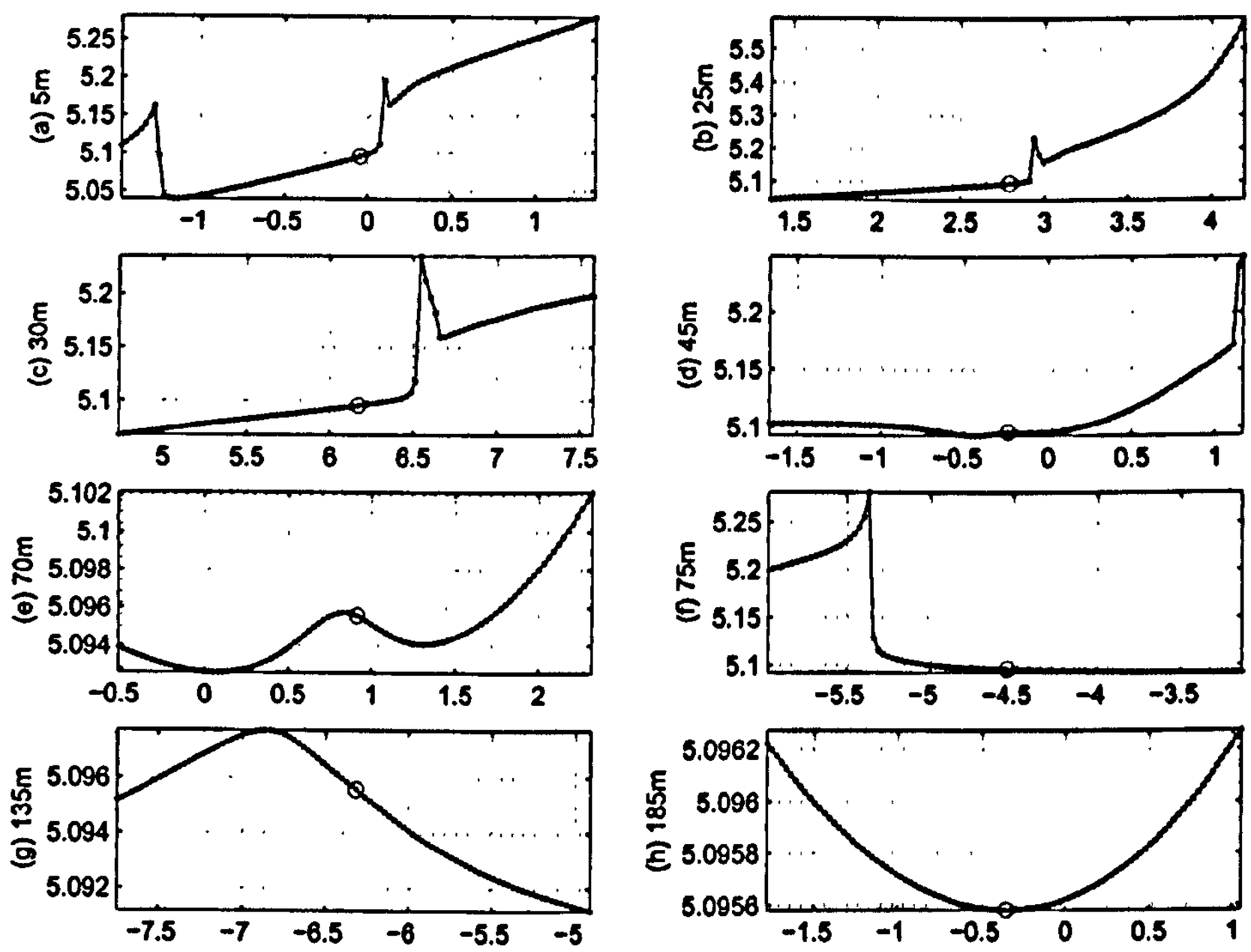


Figure 3.13: Effect of individual Steer control point values on Objective Function

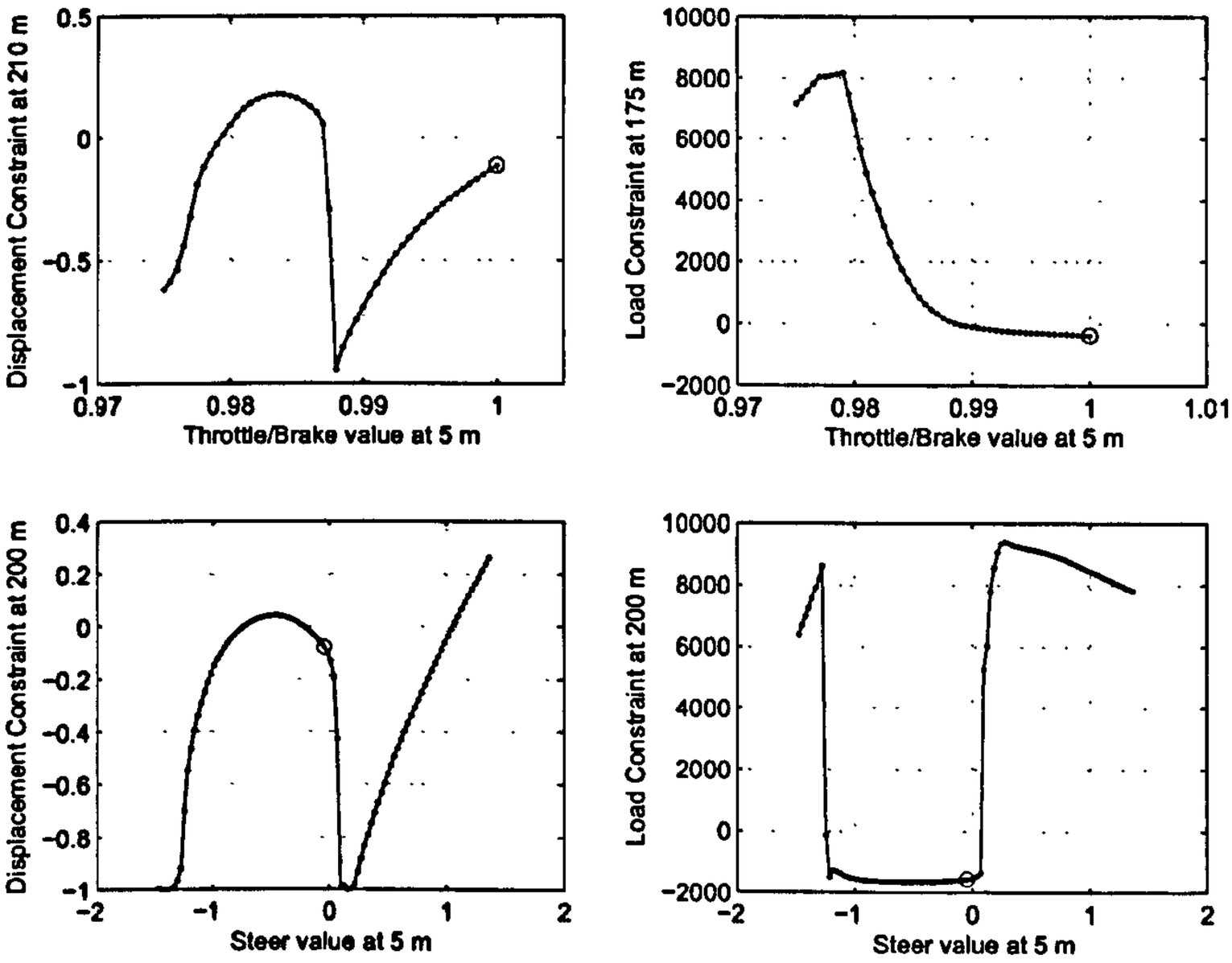


Figure 3.14: Steer and Throttle Effect on Constraints

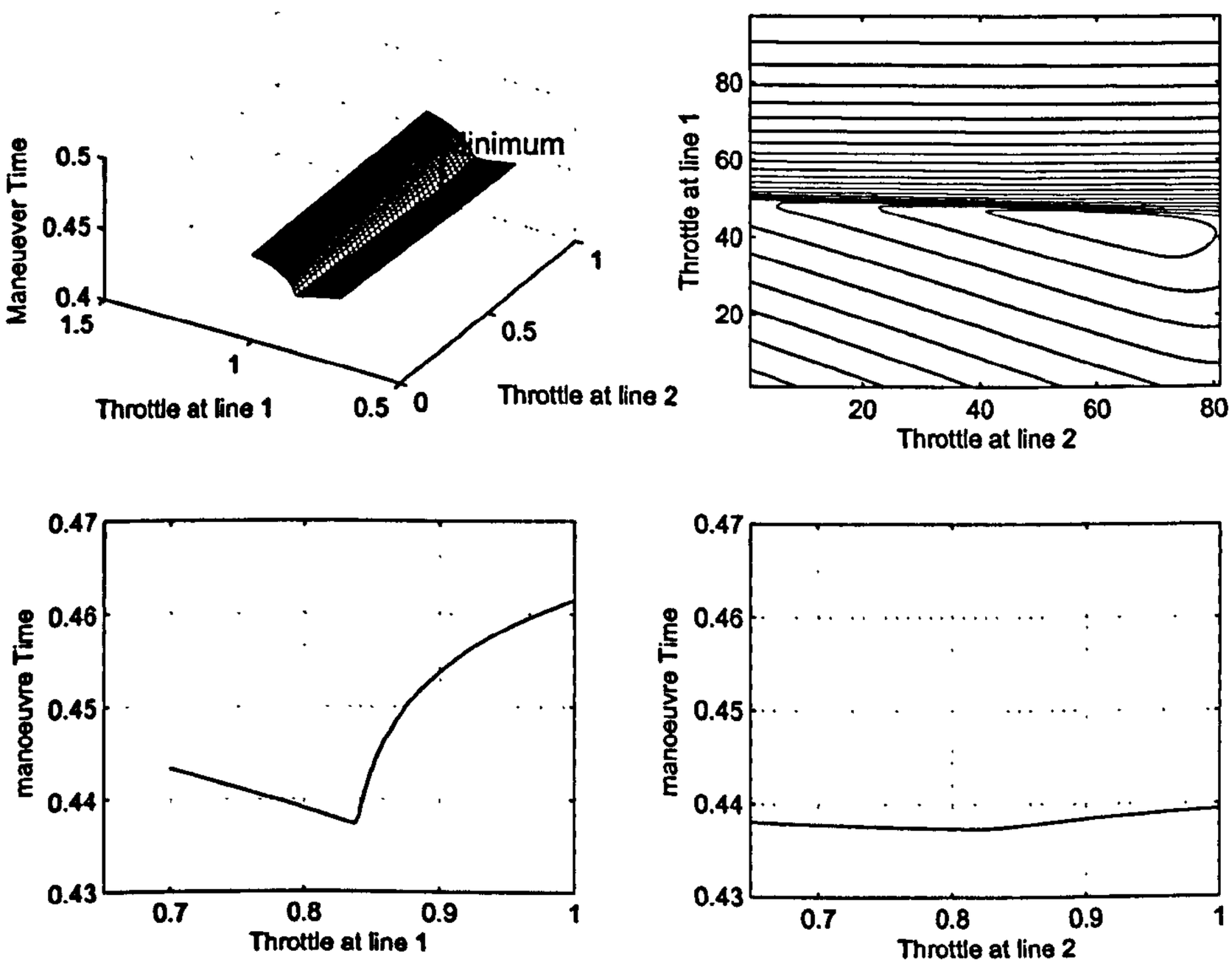


Figure 3.15: Objective function for a simple problem with two throttle control points

3.4 Influence of controller sample rate on manoeuvre time

In this section we show the effect of controller bandwidth on manoeuvre time. Figure 3.16 shows a plot of manoeuvre time against control spacing for the distance based method. The control spacing is varied from 20m to 1m in steps of 1m. It is interesting to note that time gains are still made as the control spacing is increased from 2m to 1m. Figure 3.17 shows a similar plot for the time based method. This curve becomes flat at around the 15Hz point. A choice of controller discretization interval that is representative of a human driver is considered in section 3.6.

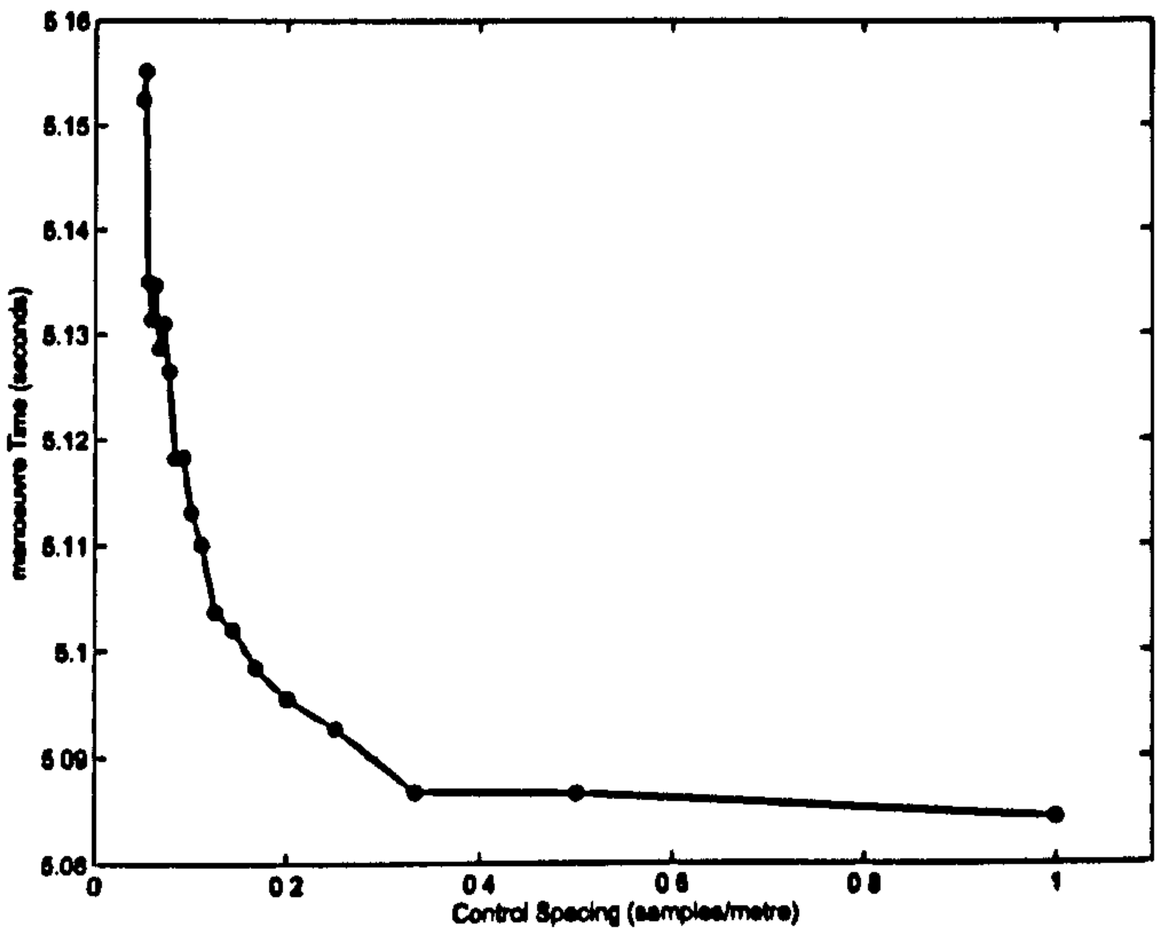


Figure 3.16: Sensitivity of Optimal manoeuvre Time to Control Spacing (distance method)

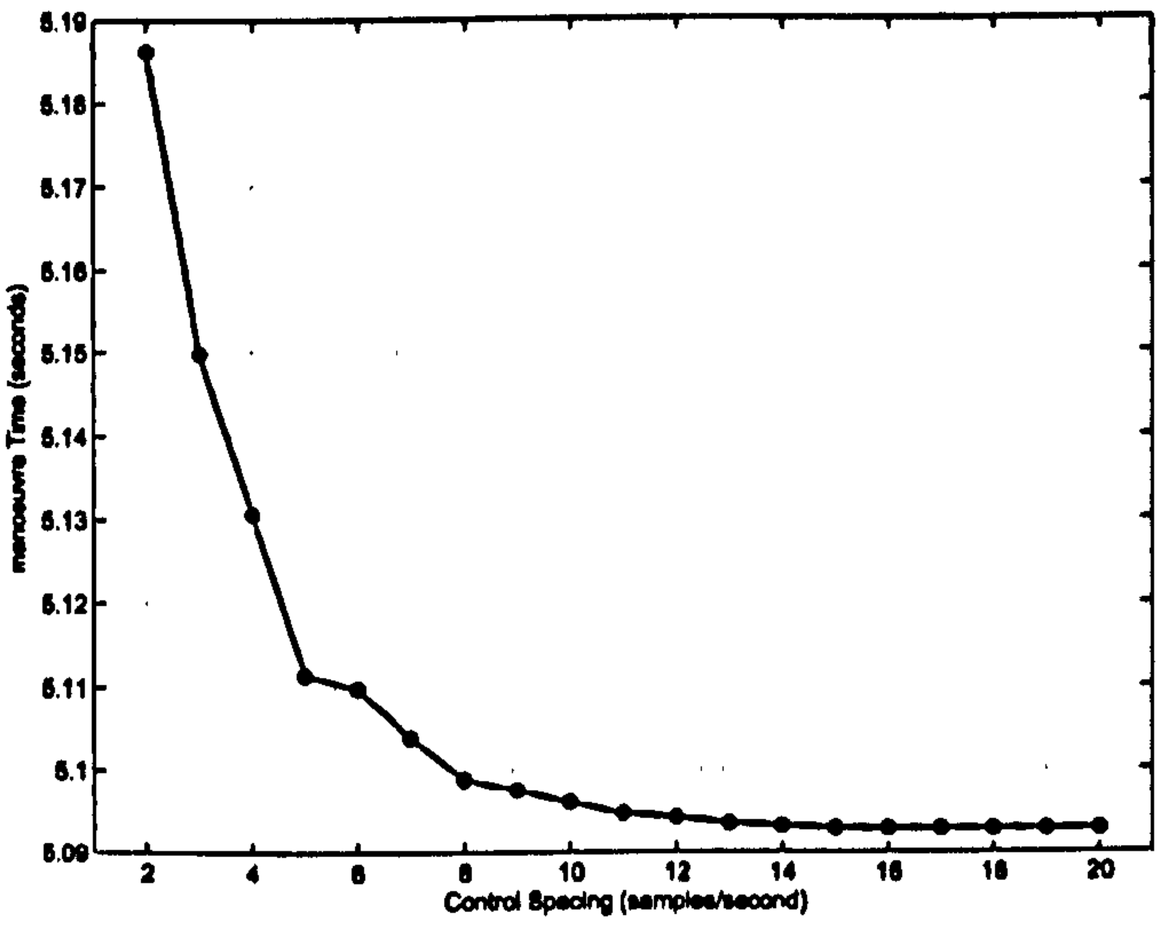


Figure 3.17: Sensitivity of Optimal manoeuvre Time to Control Spacing (time method)

3.4.1 Comparison of 1m, 5m and 10m control spacings

Figure 3.18 shows the throttle/brake and steer values for controls spacings of 1,5 and 10 metres. The refinement in the controls can clearly be seen as the discretization interval is reduced.

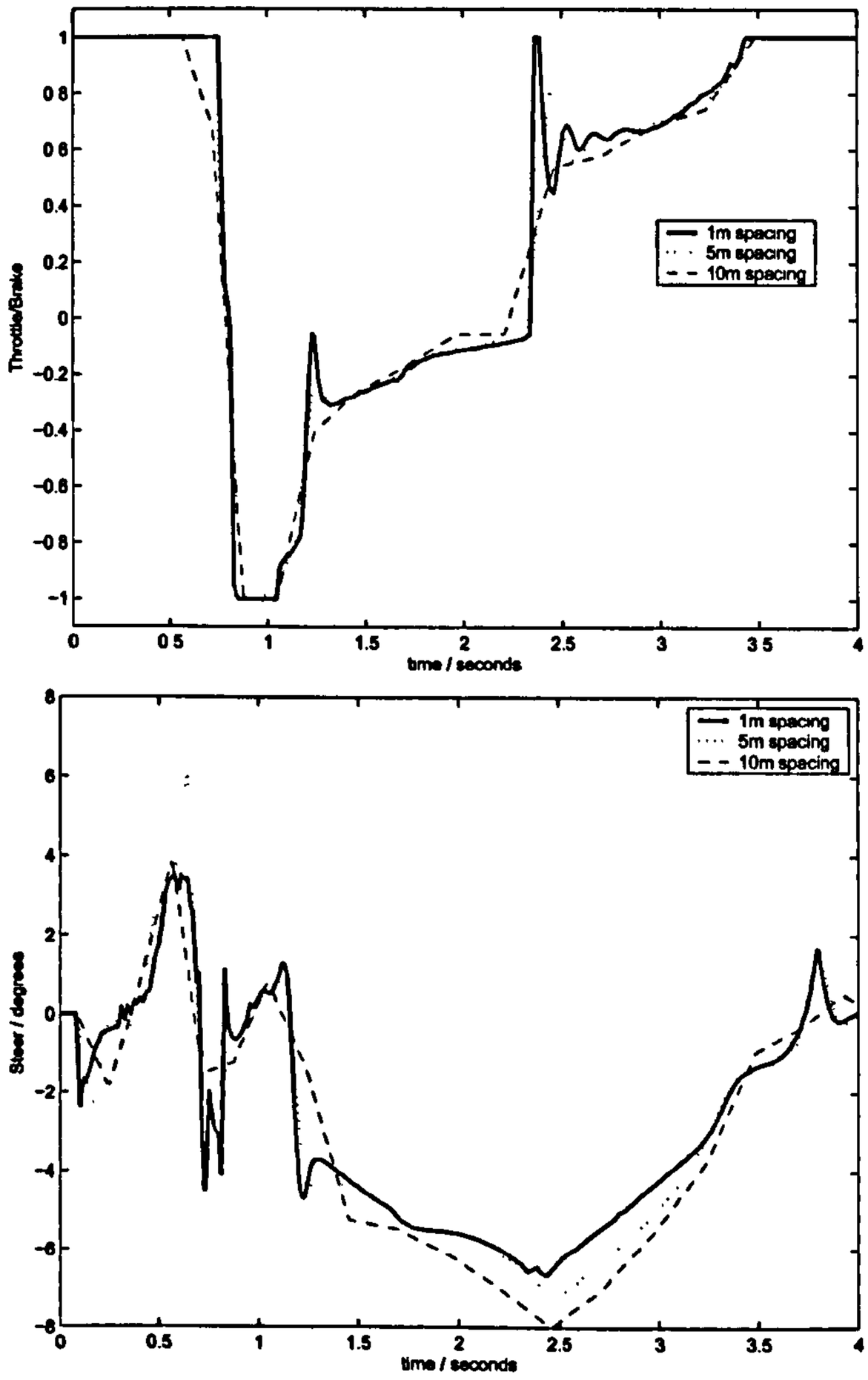
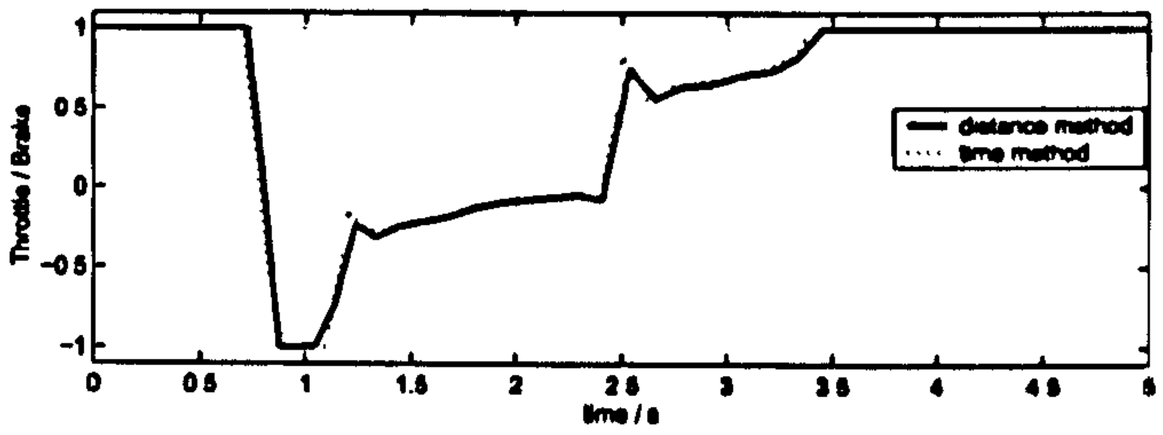


Figure 3.18: Comparison of Throttle/Brake and Steer values for 1m, 5m and 10m control spacings

3.4.2 Comparison of time and distance based Solutions

Figure 3.19 gives a comparison of the solutions from the time and distance based method.



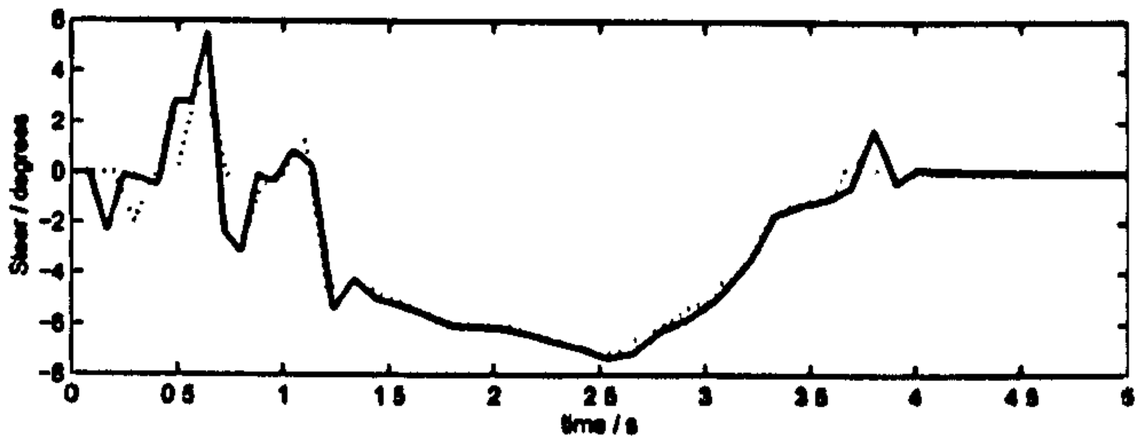


Figure 3.19: Comparison of 5m distance method and 10Hz time method

3.5 Validation of the limit nature of the solution

In this section we attempt to establish that the computed optimal solution 1 from table 3.5 is representative of the limit behaviour of the vehicle.

3.5.1 Wheel Speeds, Loads and Tyre Efficiency

The generation of tyre efficiency diagrams is described in Appendix B. Figures 3.5.1 and 3.20 show the tyre efficiency diagrams for the vehicle state trajectory associated with the optimal solution. The plots show that the tyre useage for all four tyres is close to 100% efficiency during all tyre limited parts of the manoeuvre (in the initial and final parts of the manoeuvre the vehicle performance is limited by available engine torque). The effect of the initial steer and counter-steer behaviour is shown very clearly in the tyre efficiency and wheel load plots. First the left then the right front tyre become fully unloaded indicating that 100% of the available load transfer at the front axle has been used. Both front tyres reach 100% efficiency during this stage of the manoeuvre. During corner exit, the right front tyre becomes fully unloaded for a period of approximately 0.5s. The minimum wheel load constraint in this configuration (RH/1) is set to 1N. This constraint is therefore active at the solution. This appears to be generally true for other manoeuvres and vehicle configurations such as those studied in the following chapter.

3.5.2 Stability Derivatives

The stability derivatives are defined in Appendix E. Figure 3.5.2 and 3.23 show the yaw moment and lateral acceleration derivatives respectively. The yaw stiffness derivative is given by the change in yawing moment N resulting from a change in chassis side-slip angle β. If the change in N is such as to attempt to increase the yaw moment opposing the change in β then N_β is positive. This is often described as a 'weather vane' effect. If a weather vane is displaced at an angle to the wind, a restoring moment will be generated

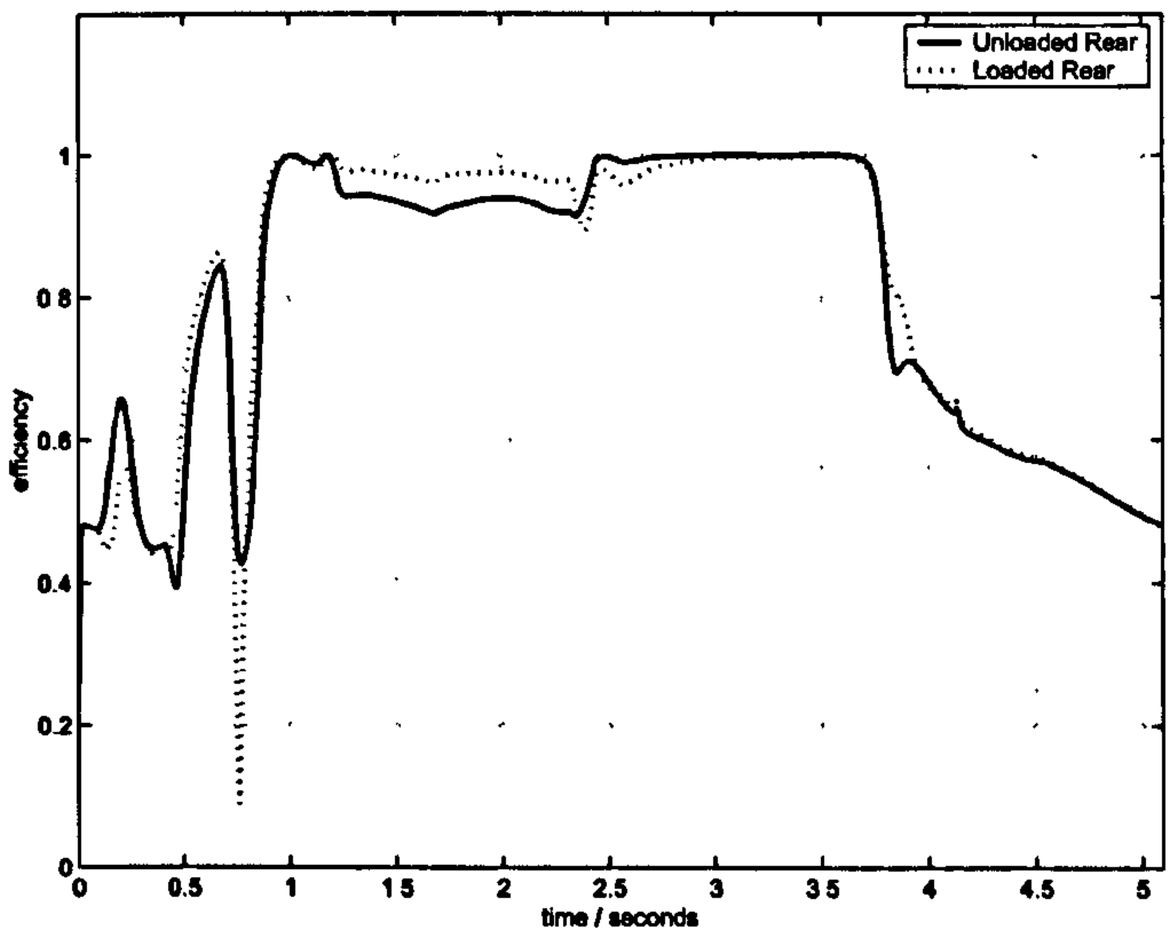


Figure 3.20: Rear tyre efficiency

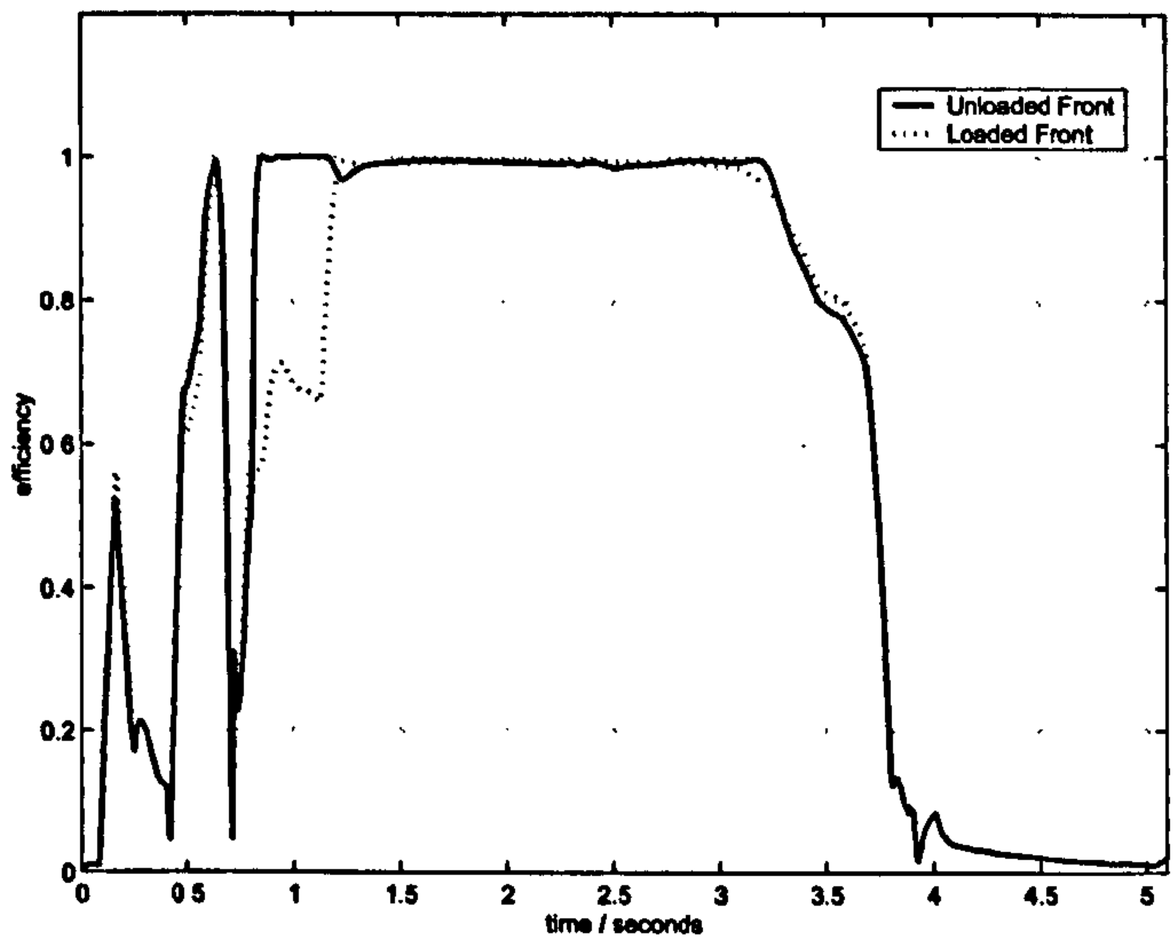


Figure 3.21: Front tyre efficiency

to attempt to reduce the angle. If N_β is negative, the vehicle is in an open-loop unstable state, that is, unless corrective control action is taken, the side-slip angle will continue to increase and the car will spin. The yaw damping derivative is the change in yawing moment N resulting from a change in yaw rate r. If an increase in yaw rate results in a decrease in the yaw moment opposing the yawing motion, the vehicle is in an open-loop unstable state. That is, unless corrective control action is taken, the yaw rate will continue to increase and the car will spin.

It is interesting that the yaw stiffness derivative is seen to go negative during heavy braking. Hence the optimized trajectory passes through a region where the vehicle response is open-loop unstable (Appendix E). The yaw damping derivative is negative throughout the manoeuvre indicating that an increase in yaw rate always results in an increase in the

opposing yaw moment. The control moment derivative N_δ indicates that from the moment the car is turned into the corner, small changes in steering have very little effect on the yaw moment. This suggests that at the limit, the driver should sense very little response from the steering. This is in agreement with the following quotation from Sir Stirling Moss [38]:

> ...once one has presented a car to a very high speed corner it's rather like throwing a dart - when it has left your hand you can't do a thing about its path. If you present a car accurately to such a corner it will track through a long drift on virtually a predestined trajectory. You can make tiny adjustments, but once you have presented it to the corner you can only adjust the trim, not make major changes of direction, not if you are on the limit.

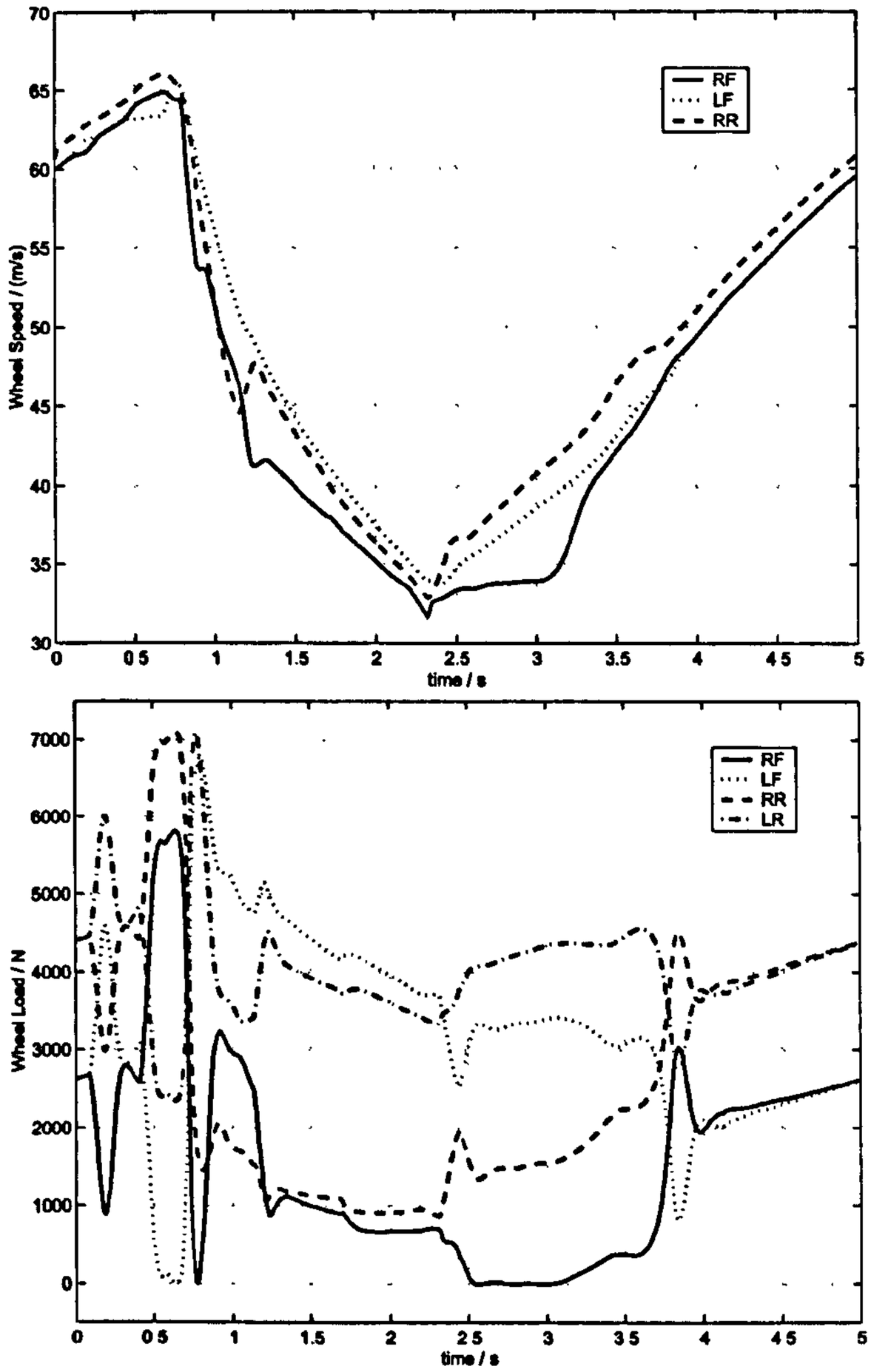


Figure 3.22: Wheel Speeds and Loads

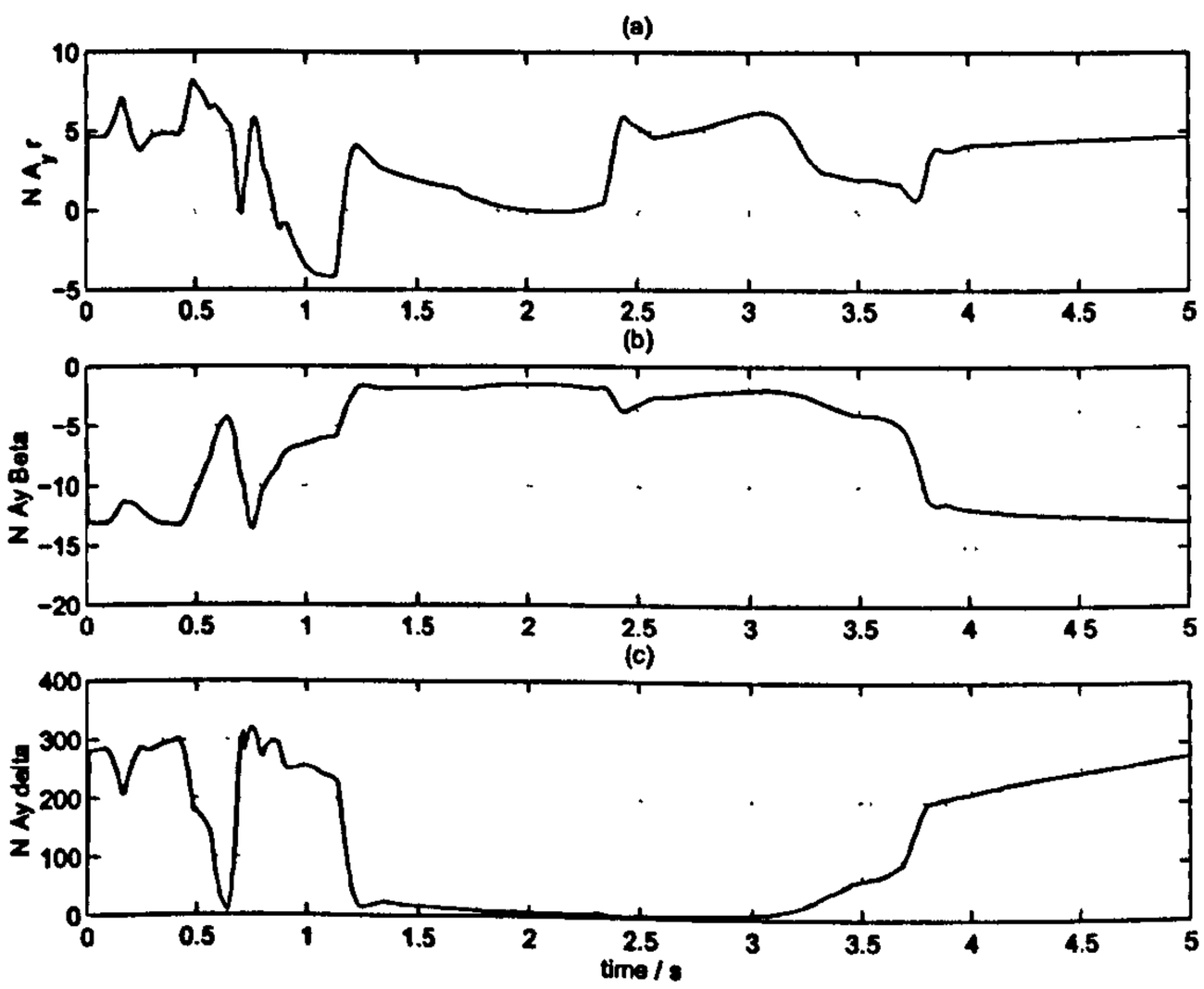


Figure 3.23: A_y Derivatives

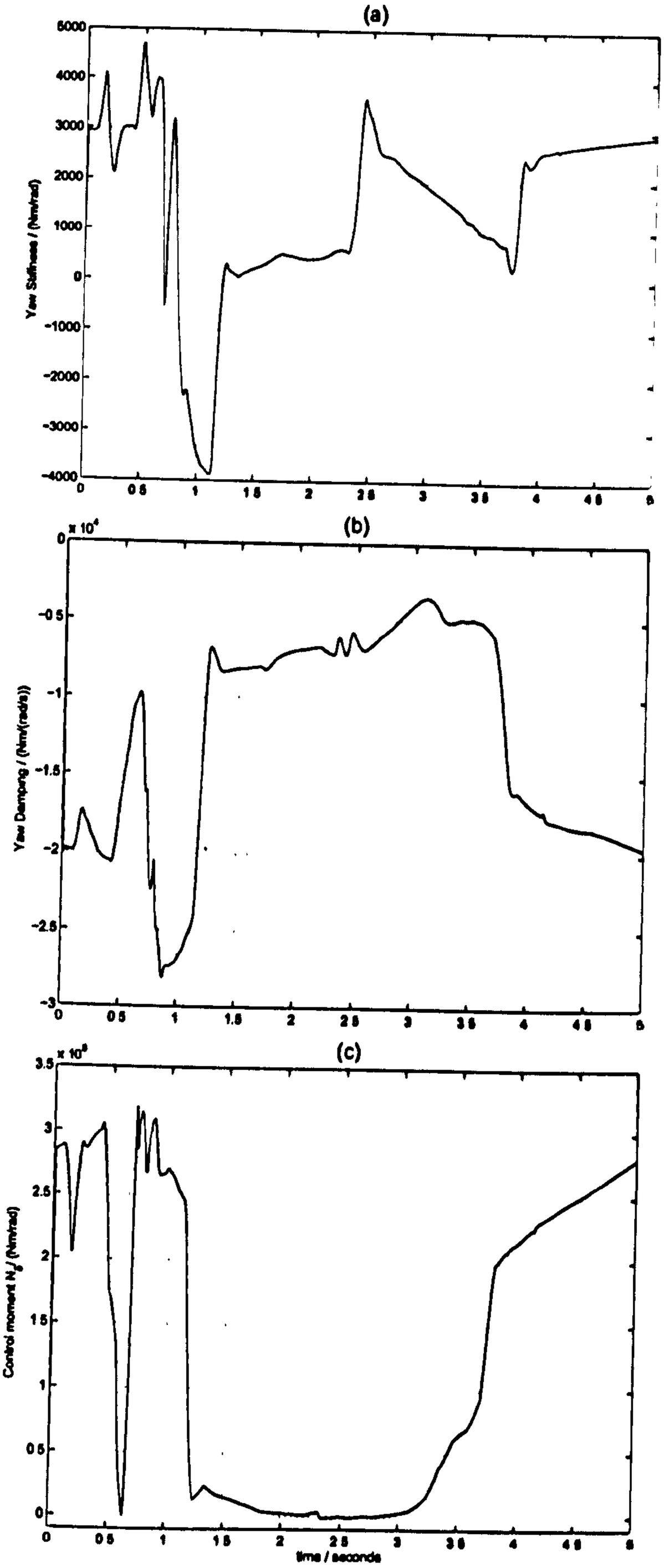


Figure 3.24: Yaw moment derivatives (a) N_β (b) N_r (c) N_δ

3.6 Consideration of the Performance limits of the expert human driver

> Why is one driver quicker than another ? He's got more confidence. Sometimes he gets it by over-riding certain fears and getting away with it
>
> Mario Andretti

It has been shown (and should be expected) that the optimizer produces state trajectories that exhibit extreme behaviour. The optimizer needs to leave no margin for error and is given as many attempts as it needs to walk slowly up to the limit of vehicle performance. A human driver has to deal with uncertainties in the environment such as variaton in tyre condition, changes in fuel load and wind strength/direction. The human driver gets a single attempt at each corner each lap. In this section we formulate hypotheses about the performance limits of the human driver. We then propose a number of modifications to and constraints on the MTM method to produce more human-like performance. It is beyond the scope of this work to study human performance in great detail. What follows is merely an effort to reduce the most obvious super-human excesses of the MTM method.

3.6.1 Human Controller Bandwidth

We have shown that improvements in manoeuvre time are still made at control sampling rates as high as 20Hz. This is considerably in excess of the frequency obtainable by a driver. It would be useful to study vehicle behaviour using the MTM method with a control discretization interval that is representative of a human driver. The literature on human-in-the-loop systems is mainly as a result of aircraft pilot modelling/analysis [27, 45, 47] and also for haptic interaction in man machine systems [2, 52]. The human is a very complex non-linear system that does not submit to simple analysis. The following is a passage from Brooks [52]:

> ... Even when the sensing and control parameters are held constant, the human operator does not have one single, overall bandwidth. For example, given the same machine, same sensory modalities, the same functional task, and the identical operator, the human response can be significantly different based on the situation. Apparently, a humans bandwidth is a funtion of the situational mode in which he is operating: specifically, a human operator has different maximum bandwidths for unexpected signals [1-2 Hz], versus periodic signals [2-5 Hz], versus internally-generated or learned trajectories [≈5 Hz], versus reflexive actions [≈10 Hz].

The human input/output system is asymmetric. The main sensing input relevant to driving is vision which has a bandwidth of approximately 50Hz [52]. The main actuating outputs relevant to driving are the arms/hands and legs/feet. The fingers have a bandwidth of approximately 5-10 Hz. The larger limbs have a bandwidth in region of 2-6 Hz [39],[52].

The elbow has a bandwidth of around 4-6 Hz in open loop motion and 2 Hz in a closed loop tracking task [39]. Current driver models separate the driver controls into an anticipatory or pre-cognitive part and a reactive feedback part [44, 7, 19, 53]. The anticipatory part is as a result of learnt behaviour and can be considered as a driver developed internal model of the vehicle dynamics. As suggested above, this represents the higher bandwidth part of the control.

Frequency response analysis is fundamentally suited to the analysis of the steady state response of linear systems. The human is clearly non-linear and furthermore we are interested in modelling transient behaviour. A compromise must be found to approximate the human as a sampled data system with a fixed temporal or spatial sampling rate.

Considering first the time based method. In the data sampling problem the Nyquist criterion states that a signal must be sampled at a rate at least twice the highest frequency present in order to avoid aliasing. We face the inverse of the problem, that is synthesizing a continuous signal from a discrete one using linear interpolation. In the synthesis problem we introduce signal energy at frequencies in excess of the Nyquist frequency. Hence a sample rate of 2Hz can produce significant spectral energy above 1Hz.

The distance based method brings the additional problem that if a fixed control spacing is used, the temporal controller sampling rate will increase with vehicle speed. This is probably quite unrepresentative of human behaviour.

It should also be noted that for any given manoeuvre, there may typically be an impact on the manoeuvre time caused by the phasing of the control discretization points relative to the vehicle position along the manoeuvre. That is, it is possible that a lower sample rate may produce an improved manoeuvre time if the controller samples are positioned such that the ideal braking or turn-in point can be more accurately located. This phenomenon explains the non-monotonic behaviour seen in figure 3.16.

As a compromise we carry out most of the studies in this thesis with a sample rate of 5 to 7Hz for the time based method and 5 to 10 metres for the distance based method. This seems to produce control waveforms that are representative of professional formula 1 drivers.

3.6.2 Steer behaviour on entry

The optimized solution exhibits extreme steer behaviour on entry. A rally driver style 'pendulum turn' approach is used. During this procedure, the front tyre efficiency reaches 100%, the control moment drops to zero and full load transfer occurs at the front axle. Whilst this behaviour is convincingly optimal, it is very unrepresentative of human driver. We can eliminate this behaviour from the MTM method by eliminating the option to steer so far in advance of the corner (steer values are held at zero before this point). Examples of this are shown later.

3.6.3 Vehicle Stability, Control Moment derivative and Minimum wheel load

The largest departure from human like performance occurs during braking. The optimized rear tyre slip ratios are significantly higher than those typically achieved by a driver. This longitudinal exploitation on the rear tyres causes a lack of rear lateral support resulting in

the negative yaw stiffness derivative seen during braking. In addition to this, the steering control derivative drops to near zero from the point the car turns into the corner indicating that the steering input has very little margin to deal with control disturbances. We are faced with the following questions:

- Does a driver push the car into open-loop unstable regions (e.g negative yaw stiffness) when braking ?
- Does a driver typically retain some steering control moment ?
- How does the driver sense that the car is at the limit ?

It would be possible to reduce the rear tyre slip ratios during braking by simply applying a slip ratio contraint (this approach is used in later chapters). However, the driver is not able to directly sense slip ratio or many other of the vehicle state variables. For feedback to the closed loop part of the driver response it is likely that the driver is sensing at least the following using optical preview and by the inner ear.

- The yaw acceleration
- The sideslip angle β and its rate of change $\dot{\beta}$.
- The yaw rate of the vehicle compared with the yaw rate required to maintain the desired vehicle path at the desired speed profile.
- The response between steering angle and yaw acceleration.

The vehicle sideslip angle and yaw responses sensed by the driver are compactly represented by the yaw stiffness derivative N_β and yaw damping derivative N_r as discussed above and defined in Appendix A. We consider below the effect of placing additional constraints on the MTM method on N_β and N_δ to remove open-loop unstable regions of operation from the solution and to enforce a minimum control moment.

3.6.4 Yaw Stiffness Derivative Constraint

Here we add an additional constraint to the MTM method to enforce that the yaw stiffness derivative remains positive throughout the manoeuvre. We use set-up QSS/A with tyre set A and experimental configuration RH/1. Figure 3.25 shows a comparison of the throttle pedal position and total braking torque for the cases with and without the yaw stiffness derivative constraint. In order to maintain $N_\beta > 0$ the brake force in the early part of the manoeuvre has been reduced by approximately 30%. As a result of the constraint, the manoeuvre time increases from 5.0956 s to 5.1231 s an increase of 0.54%. Figure 3.26 gives a comparison of the loaded rear tyre slip ratio. This clearly shows that the increased stability is achieved through a considerable reduction in rear tyre slip ratio.

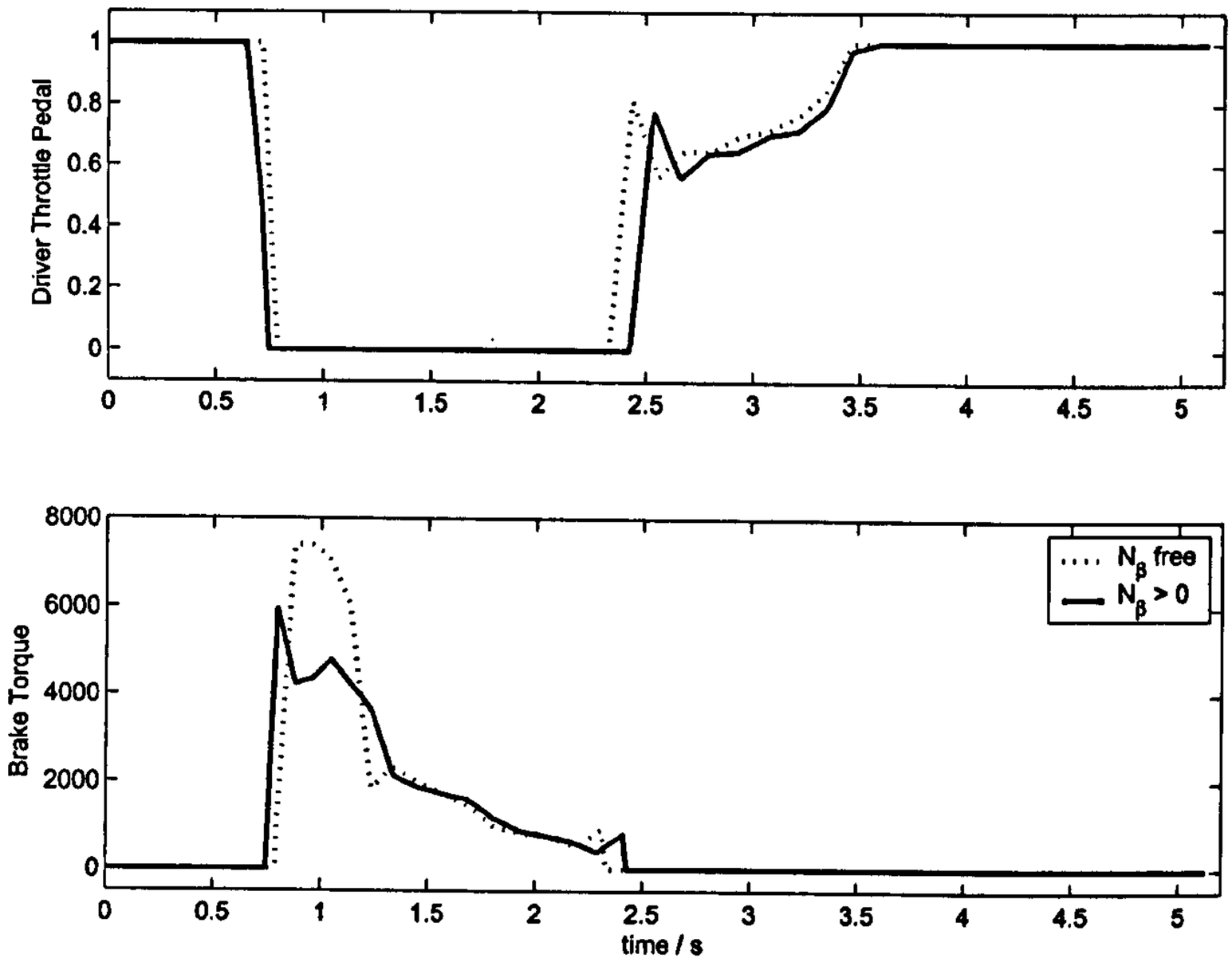


Figure 3.25: Comparison of Throttle and Brake with and without Yaw Stiffnes Derivative Constraint

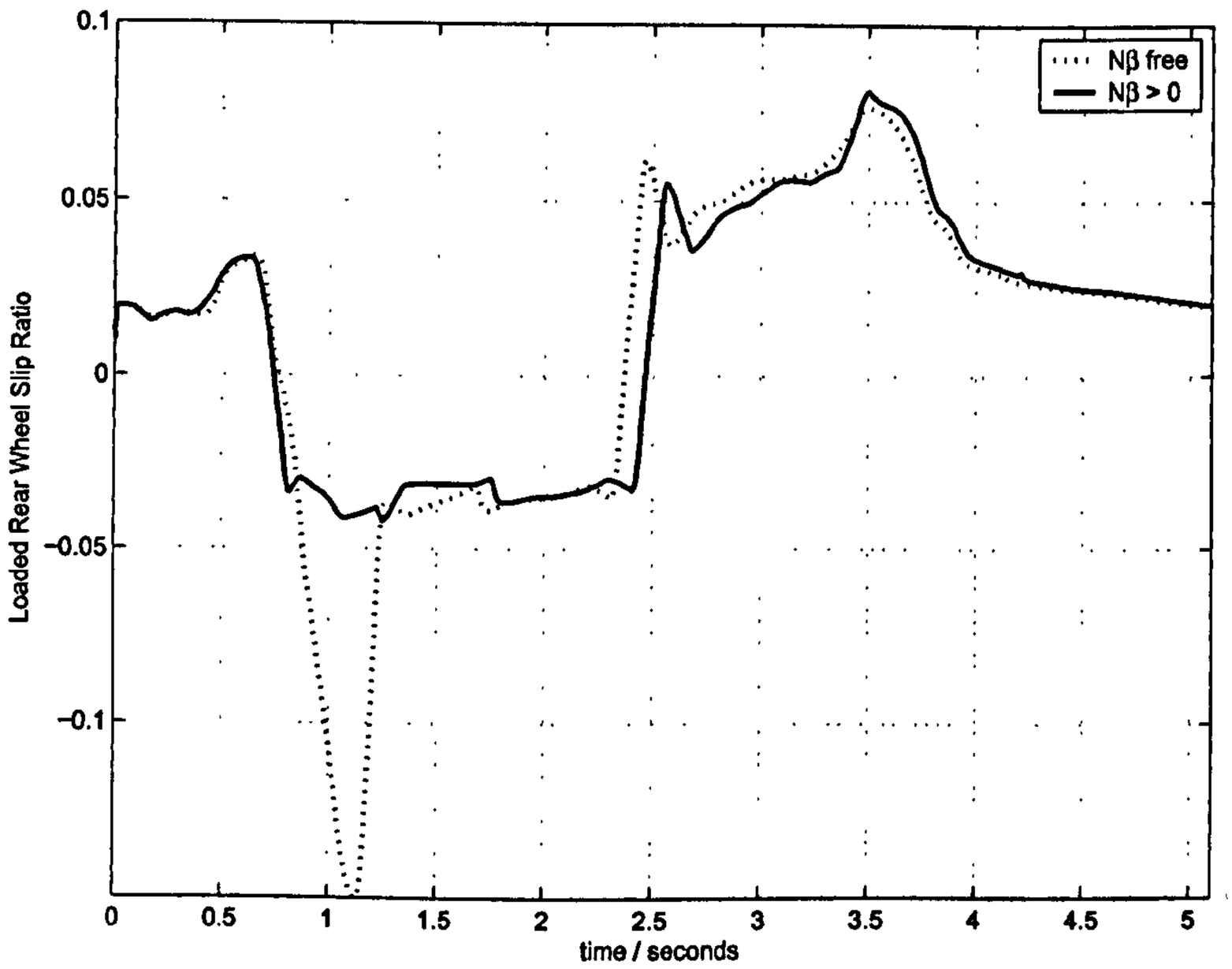


Figure 3.26: Comparison of Loaded Rear Tyre Slip Ratio with and without Yaw Stiffnes Derivative Constraint

3.6.5 Control Moment Derivative Constraint

Here we add an additional constraint to the MTM method to enforce that the control moment derivative remains above a threshold throughout the manoeuvre. When travelling in a straight line at 60 m/s the N_δ is approximately equal to 300000 Nm/rad. We set the constraint such that $N_\delta \geq 20000$. The test is carried out using set-up QSS/B with tyre set B and configuration RH/2. Figure 3.27 shows a comparison of the vehicle controls for constrained and unconstrained cases. In the constrained case, the peak steer angle is reduced by 2.5° or approximately 30%. The unconstrained solution is able to apply the power earlier in corner exit and this account for much of the time difference. As a result of the constraint, the manoeuvre time increases from 5.410 s to 5.451 s an increase of 42ms or 0.76%.

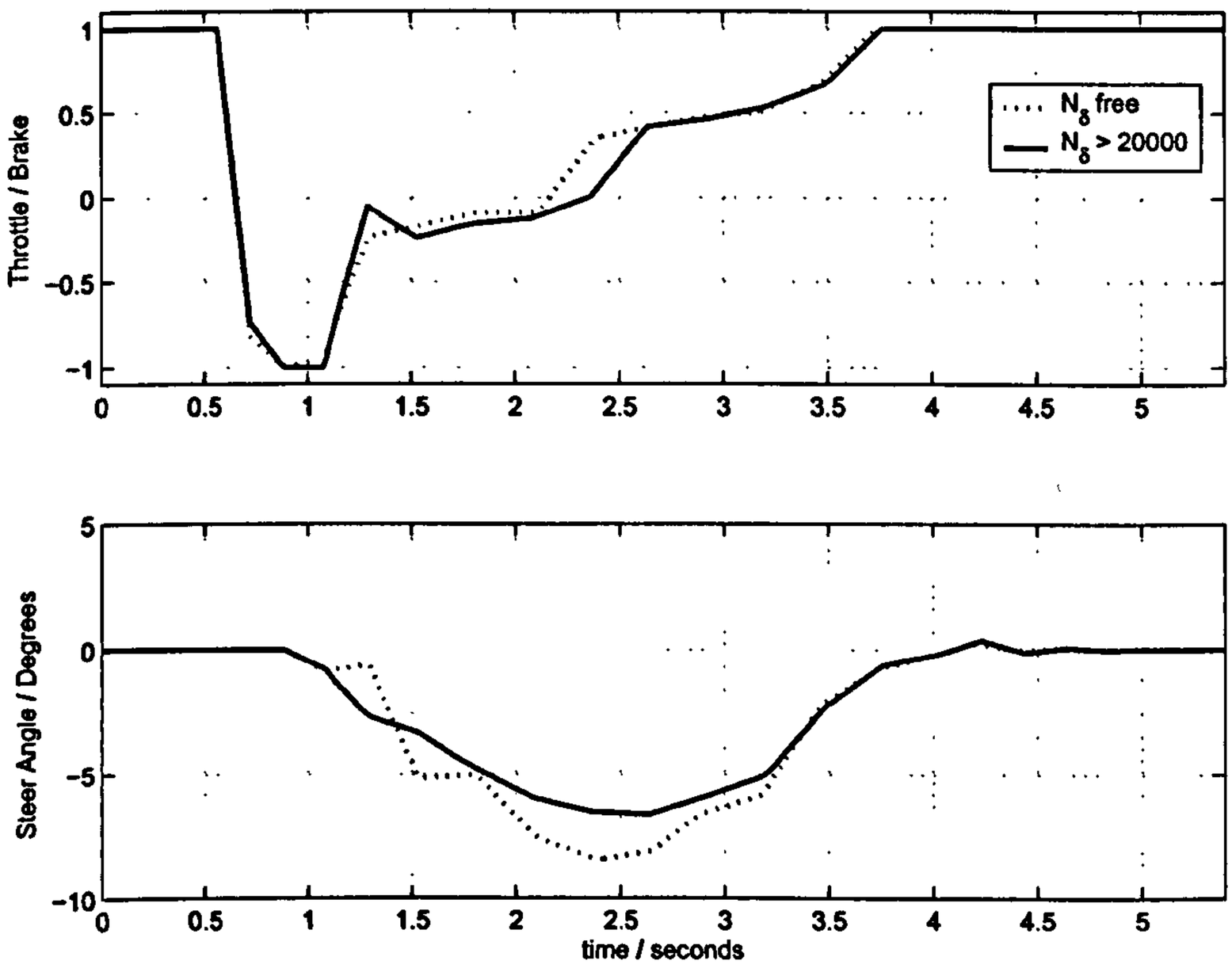


Figure 3.27: Comparison of Controls with and without Control moment derivative Constraint

3.7 Procedure for jumping out of a local minimum

The local minima described above are typical of those encountered in a wide range of manoeuvres. As shown in figure 5.1, a slower overall solution may in fact be faster over the first part of the manoeuvre. The slower solution however will typically have a slower exit speed. In order to force the MTM method to explore higher exit speed regions we can optionally introduce a further inequality constraint to force the final speed to exceed a given value.

Chapter 4

Influence of vehicle parameters on manoeuvre time, Driven line and Stability

I'm not sure they understand everything about racing cars. I am not sure that this is wise.

Jean Alesi

4.1 Summary

In this chapter we show the effect of various vehicle parameters on manoeuvre time, driven line and limit vehicle stability using the following three manoeuvres:

- Single 90° Corner (See chapter 2)
- High Speed Chicane (Becketts, Silverstone Grand Prix Circuit)
- Slow Chicane (Jerez)

The following parameters are considered:

- Mass
- Yaw moment of Inertia
- Roll moment distribution
- Longitudinal Centre of Gravity Location
- Differential set-up (Open, Locked and Limited Slip)

- Rear Grip Scale Factor

None of the parametric results presented below are intended to generalise to all vehicles operating under all circumstances. In general, the effect of any one vehicle parameter is usually closely coupled with others. For example, a change in longitudinal centre of gravity location alone may result in an increase in manoeuvre time. If it is accompanied with a change in aerodynamic balance and roll moment distribution this may become a decrease in manoeuvre time. In general, all optimal vehicle parameters are heavily influenced by the behaviour of the tyre. The purpose of the presented results is to show effects that cannot be studied with QSS simulation methods, that is, changes in driven line and vehicle stability. Comparison of driven line is achieved by analysis of the instantaneous path radius at intervals along the path. The radii are computed by least squares fitting of circles to sampled vehicle centre of gravity coordinates along the driven path. This is an effective method for driven line comparison as direct comparison of the driven paths can fail to show significant differences in path curvature.

Stability analysis is provided by plotting the yaw moment derivatives of the vehicle against time or distance along the driven line. The stability derivatives (Appendix A)are computed by applying small perturbations to steer angle δ, lateral velocity v_y and yaw rate r and hence represent a linearisation of the non-linear vehicle model around the optimized state trajectory. Analysis of stability derivatives requires careful interpretation. For example, a vehicle parameter change that affects say only the vehicle corner entry performance may have a significant impact on the stability of the vehicle in corner exit by facilitating a higher speed during the exit phase.

We consider below the effect of these six vehicle parameters using three vehicle and tyre configurations across 3 manoeuvres. An exhaustive presentation of this data would be too large for this thesis. The most interesting and representative cases only are shown. The solutions are generated using the distance based method with computational procedure A and a control spacing of 10m.

4.2 Additional manoeuvres, Jerez Chicane, Silverstone Becketts

Two further manoeuvres are presented below.

4.2.1 Slow Chicane, Jerez

Figure 4.2.1 shows the optimized driven line for a low speed chicane manoeuvre. This manoeuvre is a section of the Jerez de la Frontera circuit. The purpose of this manoeuvre it to test the vehicle response in a rapid change of direction at low speed. An example set of optimal controls and speed profile is shown figure 4.1.

4.2.2 High Speed Chicane, Silverstone Becketts

Figure 4.2.2 shows the optimized driven line for a high speed chicane manoeuvre. This manoeuvre is a section of the Silverstone racing circuit. The purpose of this manoeuvre

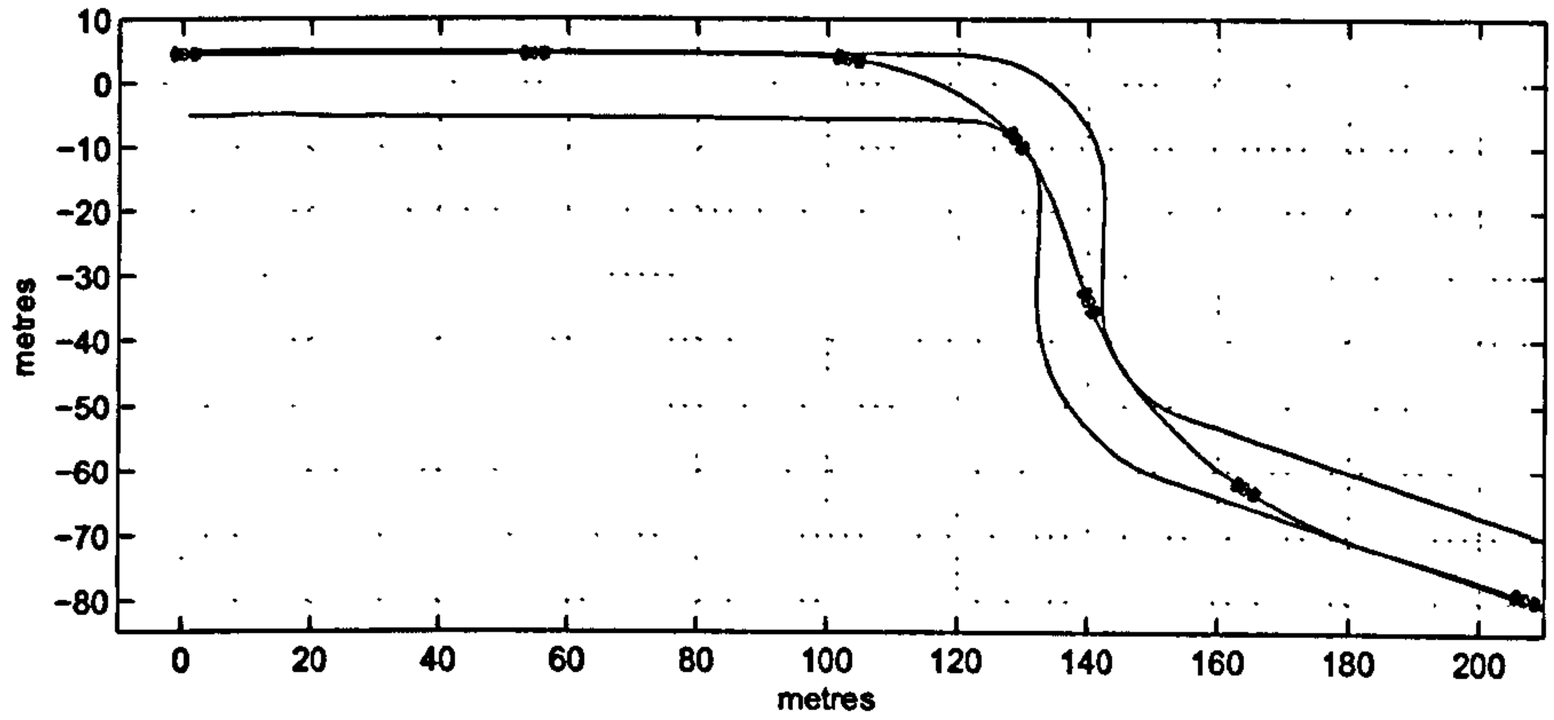


Figure 4.1: Slow Chicane driven path

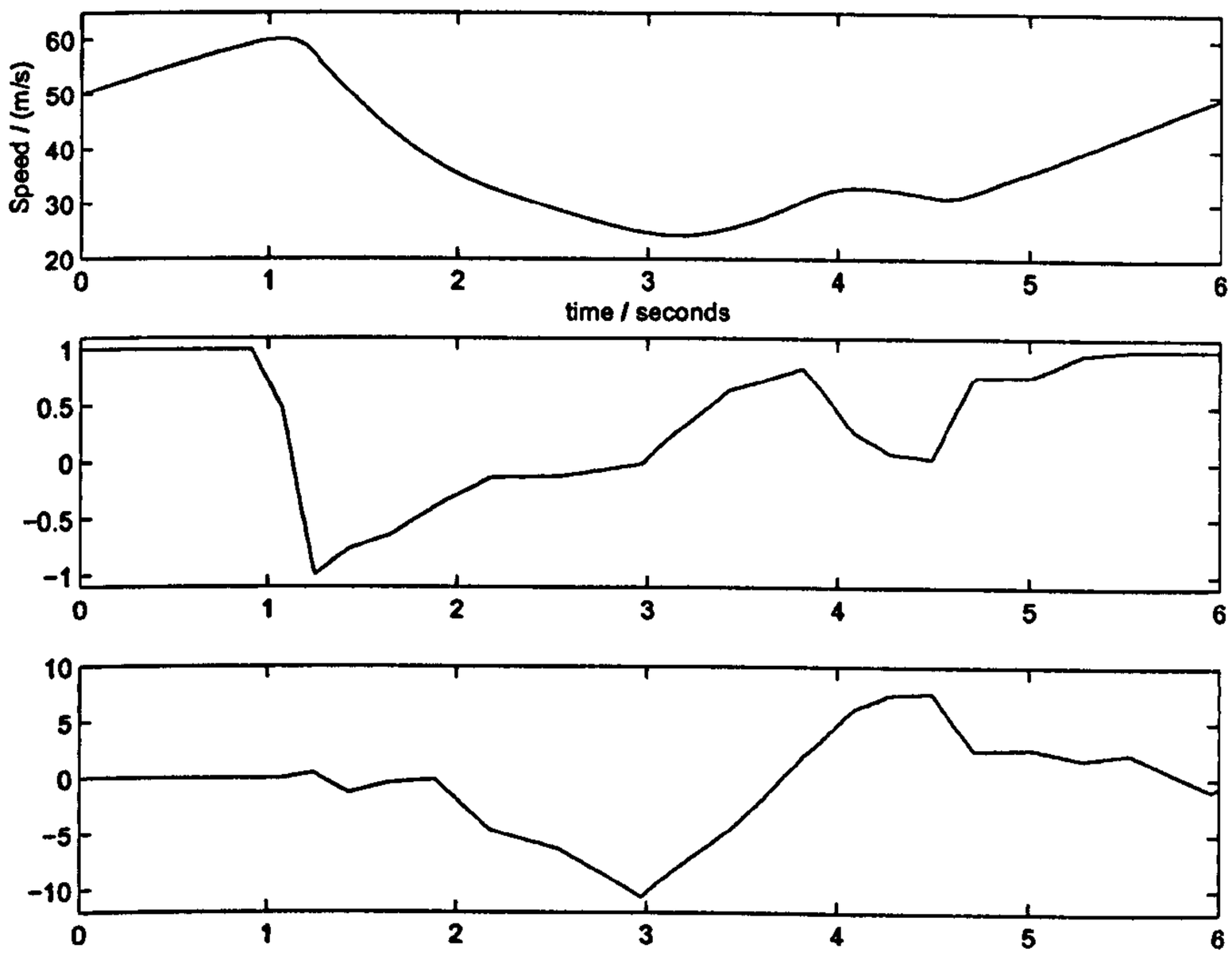


Figure 4.2: Jerez Chicane Speed profile and controls

it to test the vehicle response in a change of direction at high speed. An example set of optimal controls and speed profile is shown figure 4.3.

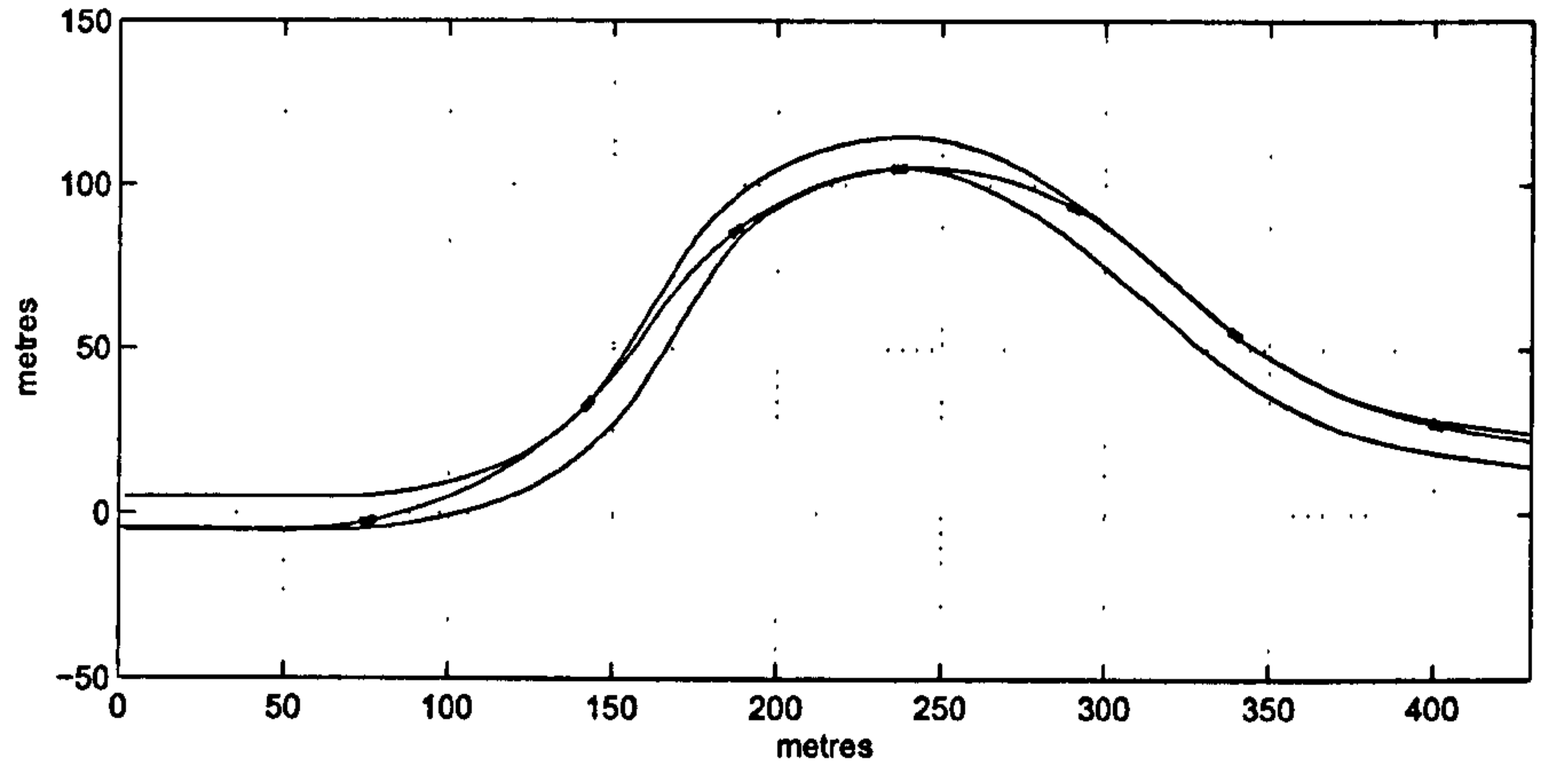


Figure 4.3: High Speed Chicane driven path

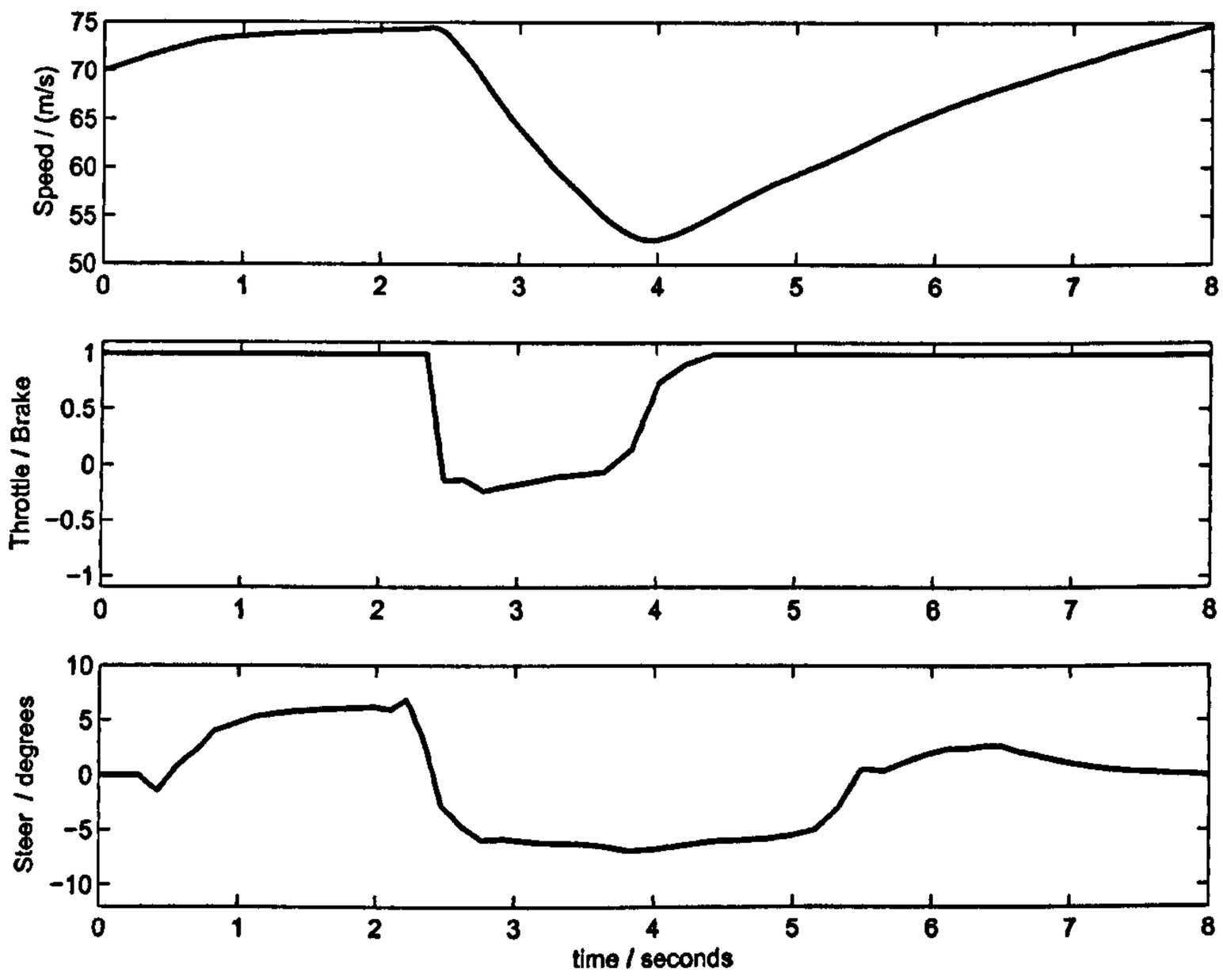


Figure 4.4: High Speed Chicane Speed profile and Controls

4.3 Mass

Here we consider the effect of vehicle mass. Mass is the most simple and well understood vehicle parameter. An increase in mass will always result in an increase in optimal manoeuvre time. For this reason, it can an also be considered as a simple means of determining the accuracy of a manoeuvre time simulator as described in section 3.3.1.2. Figures 4.5 shows the sensitivty of optimal manoeuvre time to vehicle mass for the three manoeuvres. In the single 90° corner case the two solutions described in section 3.3 can be seen clearly in the data.

manoeuvre	Time Delta (ms)	Time Delta %	Time Delta %/kg
90° corner	163.3	3.22	0.0325
Jerez Chicane	134.7	2.19	0.0221
Becketts	391.3	4.89	0.0494

Table 4.1: Mass change in manoeuvre time from 600 to 699 kg

4.3.1 Mass effect on vehicle controls

Figure 4.6 gives a comparison of the vehicle controls, understeer angle and body side-slip angle for mass values of 600 and 699 kg in the single 90° corner case. At the higher mass, as should be expected, the heavy car solution brakes earlier. The intial steering input is also slightly earlier although the main steering change occurs later. The slow speed chicane data also demonstrates two solutions.

4.3.2 Mass effect on Driven Line

The path radii of the driven lines for 600 and 699 kg are shown in figure 4.7. Turn in to the corner occurs 2-3m earlier otherwise the change in driven line is very small.

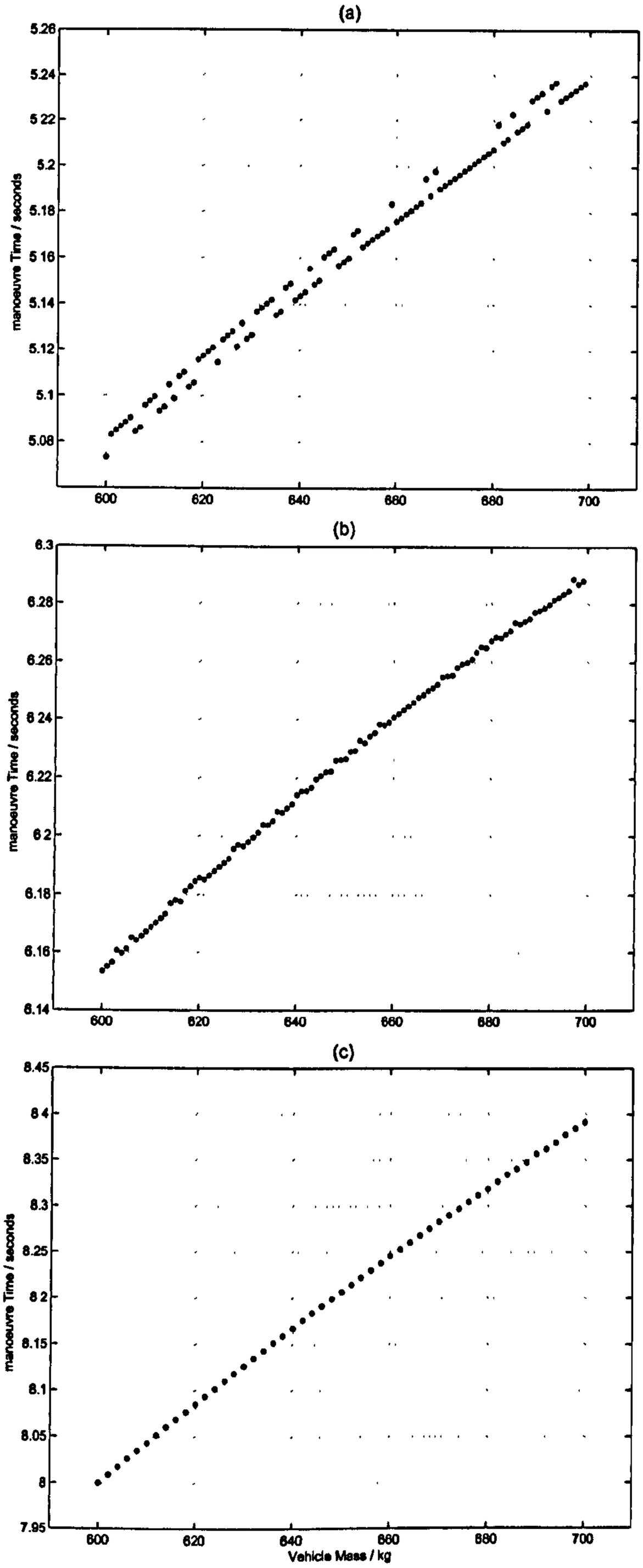


Figure 4.5: Mass effect on manoeuvre time (a) single 90° corner (b) Jerez Chicane (c) Becketts

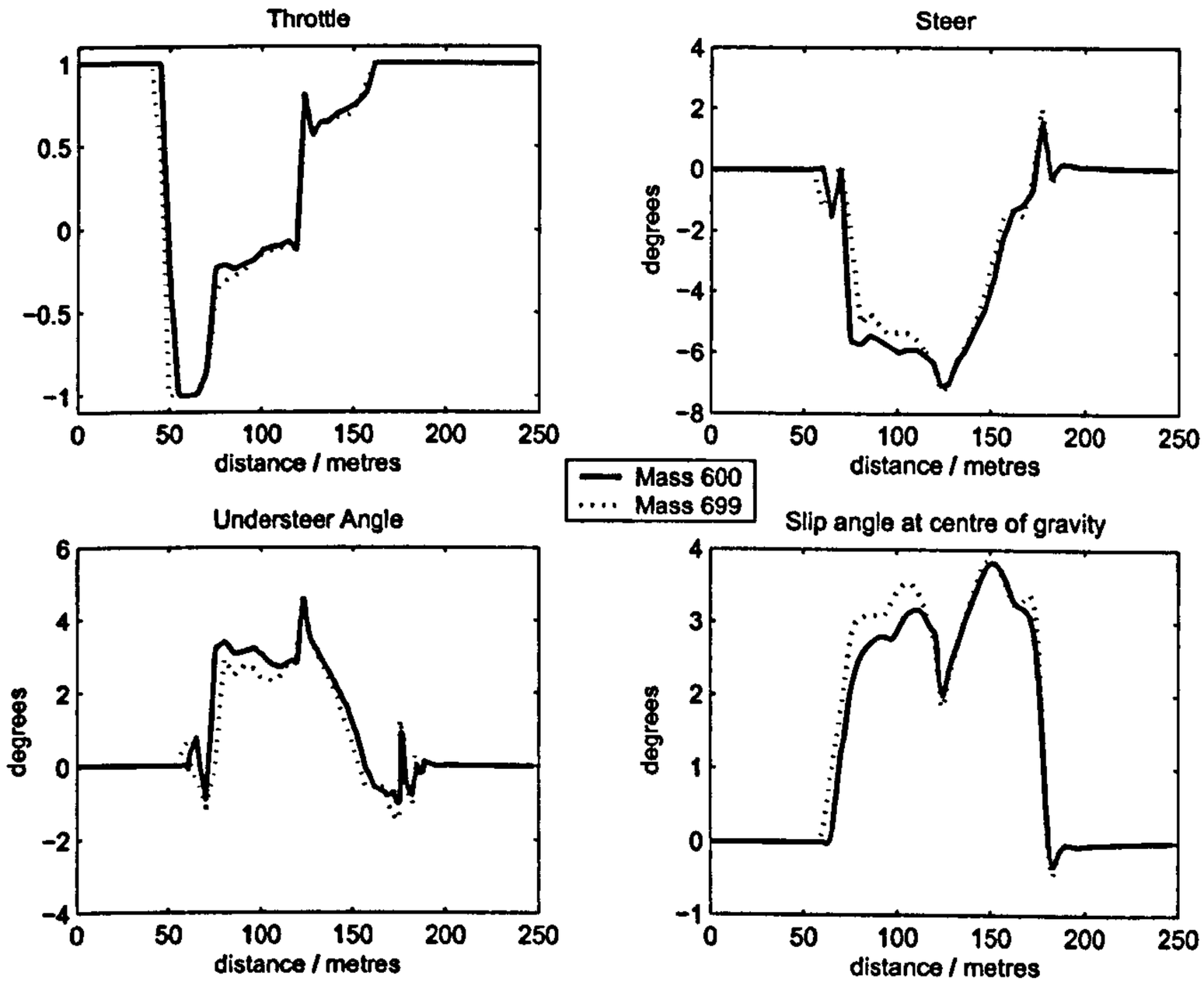


Figure 4.6: Comparison of Vehicle Mass, Single 90° Corner

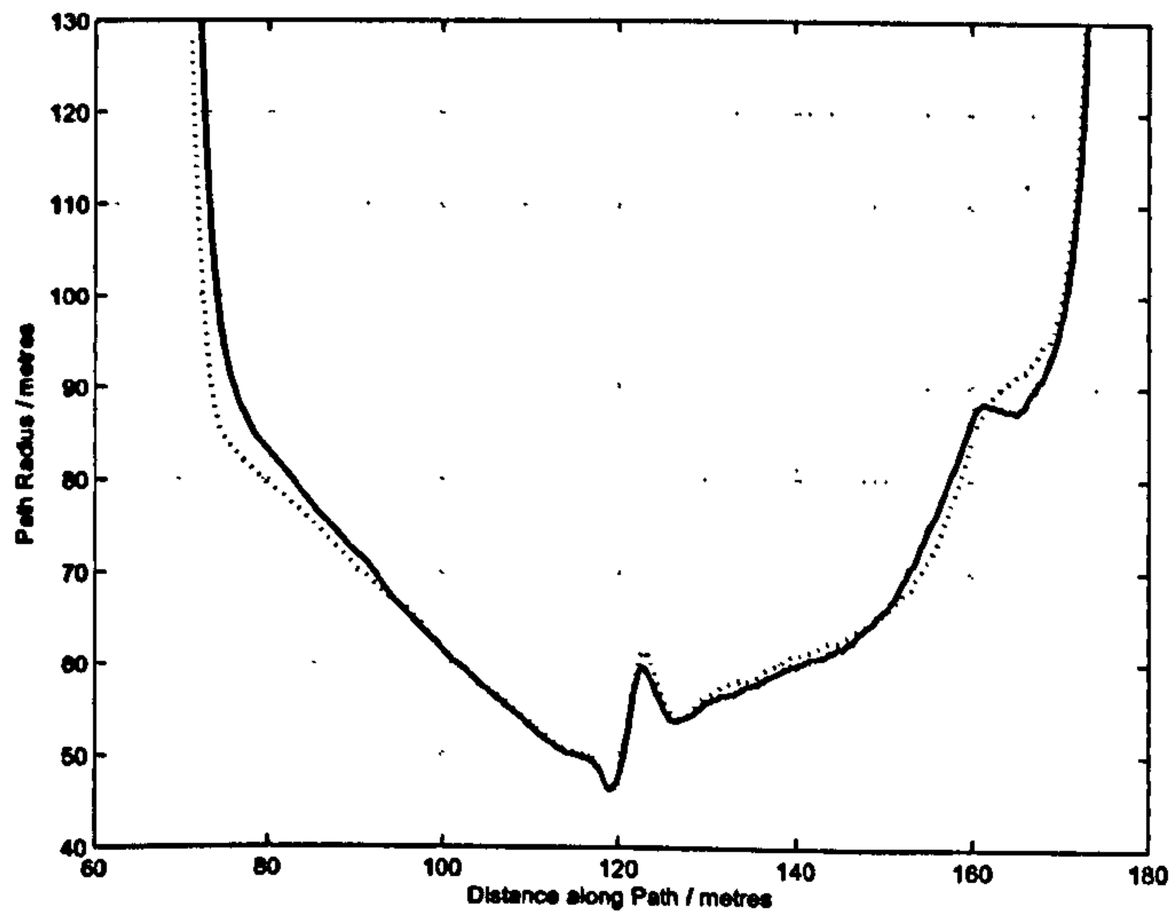


Figure 4.7: Mass effect on Driven Line, Single 90° Corner

4.4 Yaw Moment of Inertia

The effect of yaw moment of inertia on manoeuvre time was first studied by Casanova [8]. The results found here are in broad agreement with that work. Figure 4.8 shows the manoeuvre time corresponding to yaw moment of inertia in the range 400 to 1390 kgm^2. A typical modern F1 car has yaw moment of inertia in the region of 600-700 kgm^2. Over this range there is very little change in manoeuvre time for any of the manoeuvres. The most interesting data occurs in the slow speed chicane solutions. Figure 4.8(b) appears to show two solutions families visible as two arcs. The arcs cross each other at around 870 kgm^2. Each solution corresponds to a different driven line as explained below.

manoeuvre	Time Delta (ms)	Time Delta %	Time Delta $\%/kgm^2$
90° corner	6.9	0.14	0.0014
Jerez Chicane	15.3	0.25	0.0025
Becketts	5.4	0.06	0.0006

Table 4.2: Yaw inertia change in manoeuvre time from 400 to 1399 kgm^2

4.4.1 Vehicle controls

Figure 4.9 gives a comparison of the vehicle controls, understeer angle and body side-slip angle for yaw inertia values of 400 and 1380 kgm^2. At the higher inertia, the MTM method has time advanced the steering profile to compensate for the cars reduced yaw acceleration. Other than this, there is little change in the vehicle controls or in the attitude the car assumes through the corner.

4.4.2 Driven Line and Speed Profile

For the righthand turn and Becketts manoeuvres, the change in driven line with yaw inertia is minimal. Only a small effect can be seen in the righthand turn where the vehicle rapidly changes body side-slip angle as the power is applied. The most significant change in driven line and speed profile are shown in figures 4.10 and 4.11 for the slow speed chicane. The solutions to this manoeuvre consists of two groups each with a quite different driven line and speed profile. The higher yaw inertia car favours the solution that takes a tighter line through the second phase of the chicane.

4.4.3 Vehicle Stability

There is very little change in the vehicle yaw moment derivatives across the range of yaw inertias considered. It is worthing noting that the yaw acceleration resulting from a yaw moment imbalance will obviously be smaller as yaw inertia increases. Since the driver senses yaw acceleration rather than moment imbalance, the high yaw inertia car should feel more stable.

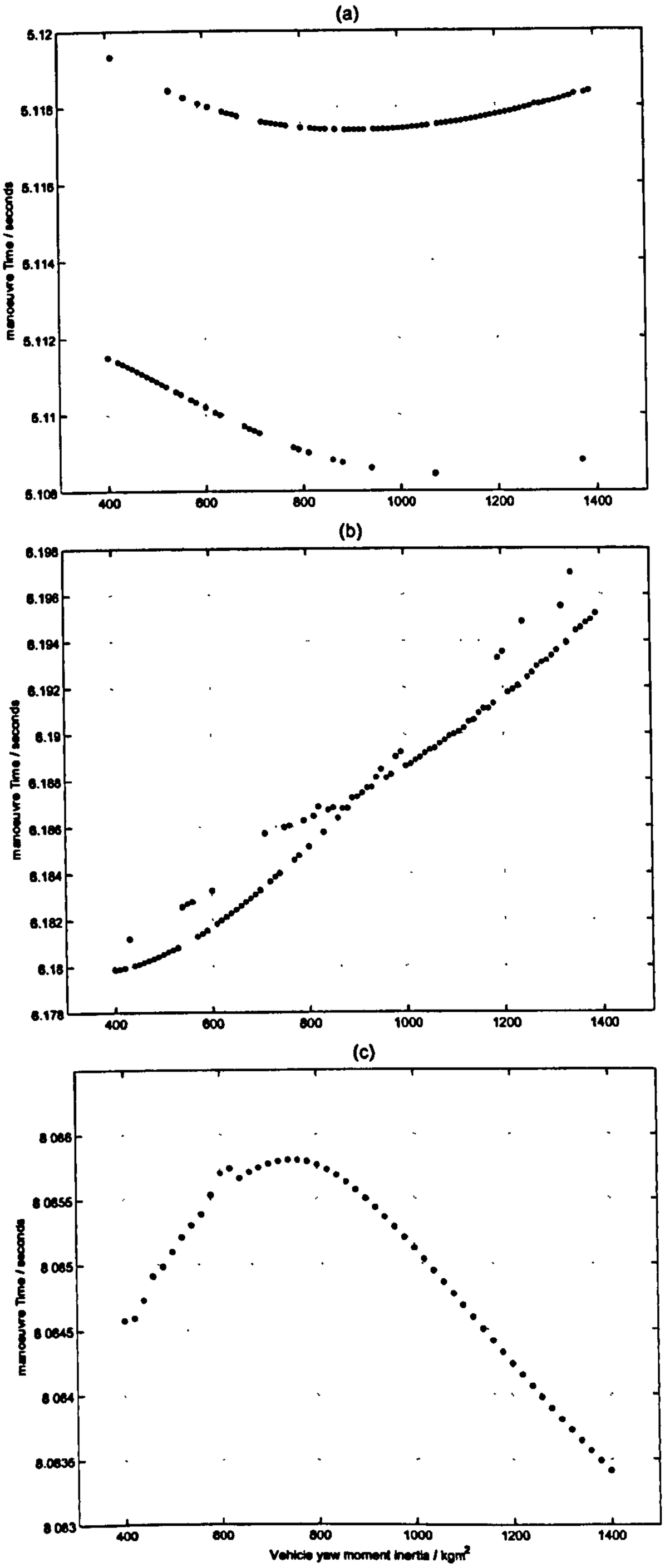


Figure 4.8: Yaw moment of Inertia effect on manoeuvre time

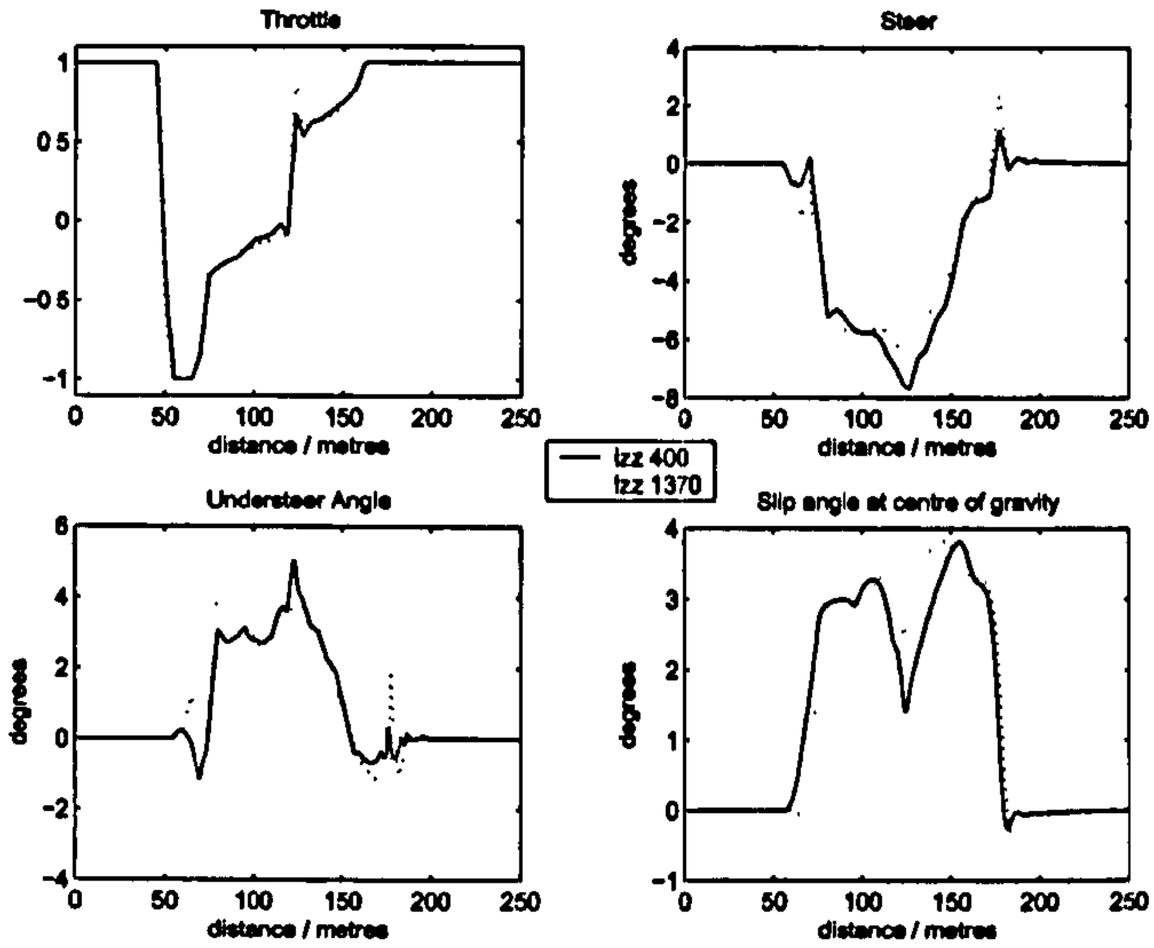


Figure 4.9: Comparison of High and Low Yaw Inertia, Righthander

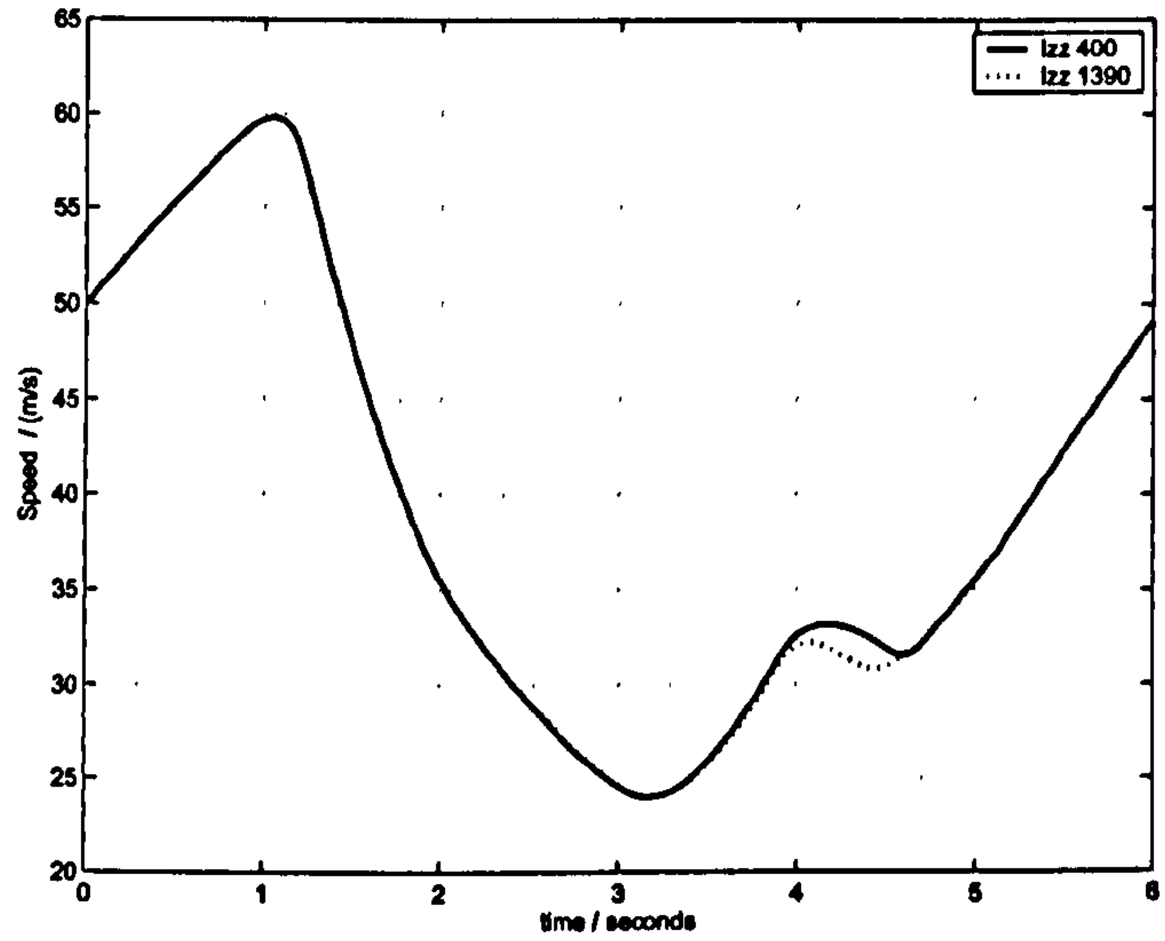


Figure 4.10: Yaw moment of Inertia effect on Speed Profile, Jerez Chicane

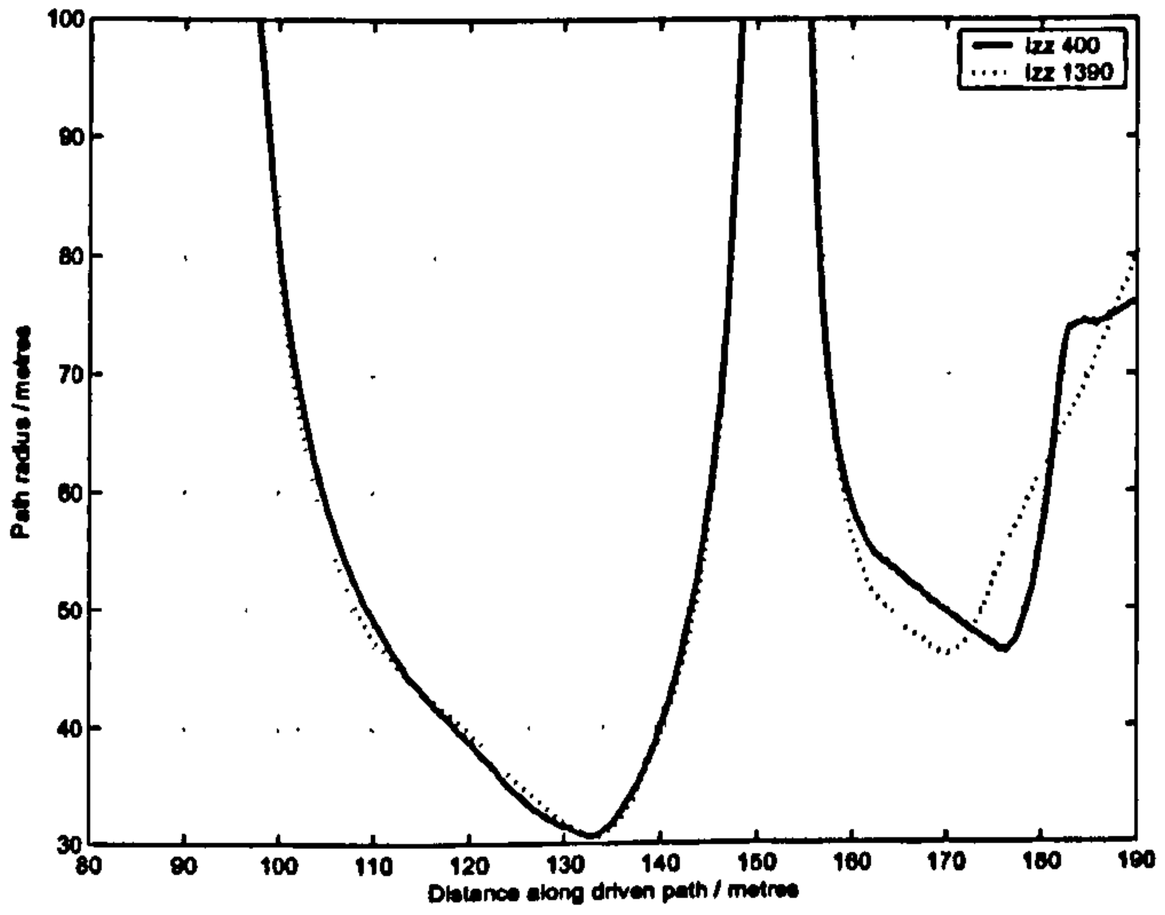


Figure 4.11: Yaw moment of Inertia effect on driven line, Jerez Chicane

4.5 Roll Moment Distribution

Roll moment distribution is one of the most readily tuneable set-up parameters on the race car. Virtually all single seater race cars are equipped with a means of rapid adjustment of this parameter through roll bars or springs. In some formulae the driver can adjust the stiffness of the front anti-rollbar from the cockpit whilst the car is on track. Moving roll moment distribution forwards is generally associated with a move towards an understeer balance and an increase in vehicle stability. Figure 4.12 shows the effect of roll moment distribution on manoeuvre time for the single 90° corner manoeuvre (using QSS Car set-up B with Tyre set 2). The manoeuvre time increases by 41ms (0.76%) across the range of roll moment distributions.

manoeuvre	Time Delta (ms)	Time Delta %	Time Delta %/%
90° corner	41.4	0.76	0.13

Table 4.3: Change in manoeuvre time for Roll moment distribution 60% to 66% Front

4.5.1 Vehicle controls

Figure 4.13 gives a comparison of the vehicle controls, understeer and body side-slip angles for the 60 and 66% roll moment cases. As expected, the forwards-roll-moment-solution shows more understeer. It also shows reduced body side-slip at the apex indicating that the rear axle is under-utilised (this is also confirmed from tyre efficiency data). The rearwards case is able to apply the power earlier in corner exit.

4.5.2 Driven Line and Speed Profile

Figures 4.14 and 4.15 show the speed and driven line for the 60 and 66% roll moment cases. The forwards-roll-moment-solution minimum speed is approximately 1.5 m/s (5.4 km/h) slower with a minimum turn radius approximately 2m lower.

4.5.3 Vehicle Stability

Figure 4.16 shows the three yaw moment derivatives for the 60 and 66% solutions. The-forwards-roll-moment-solution shows increased yaw stiffness and yaw damping delivering the expected increase in vehicle stability. Obviously, the price for the enhanced stability is paid for by the increase in manoeuvre time.

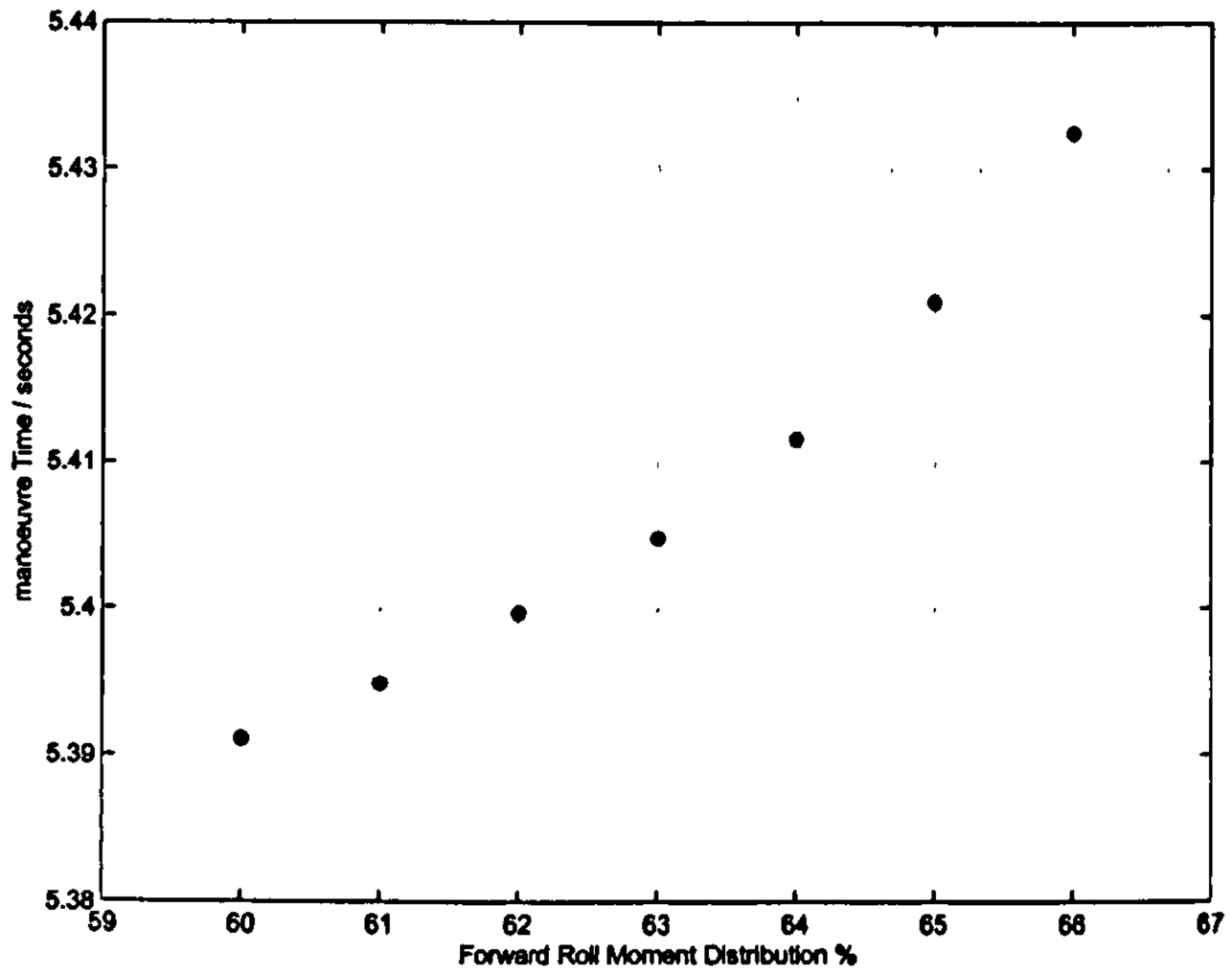


Figure 4.12: Impact of Roll Moment Distribution on manoeuvre Time - Single 90° Corner

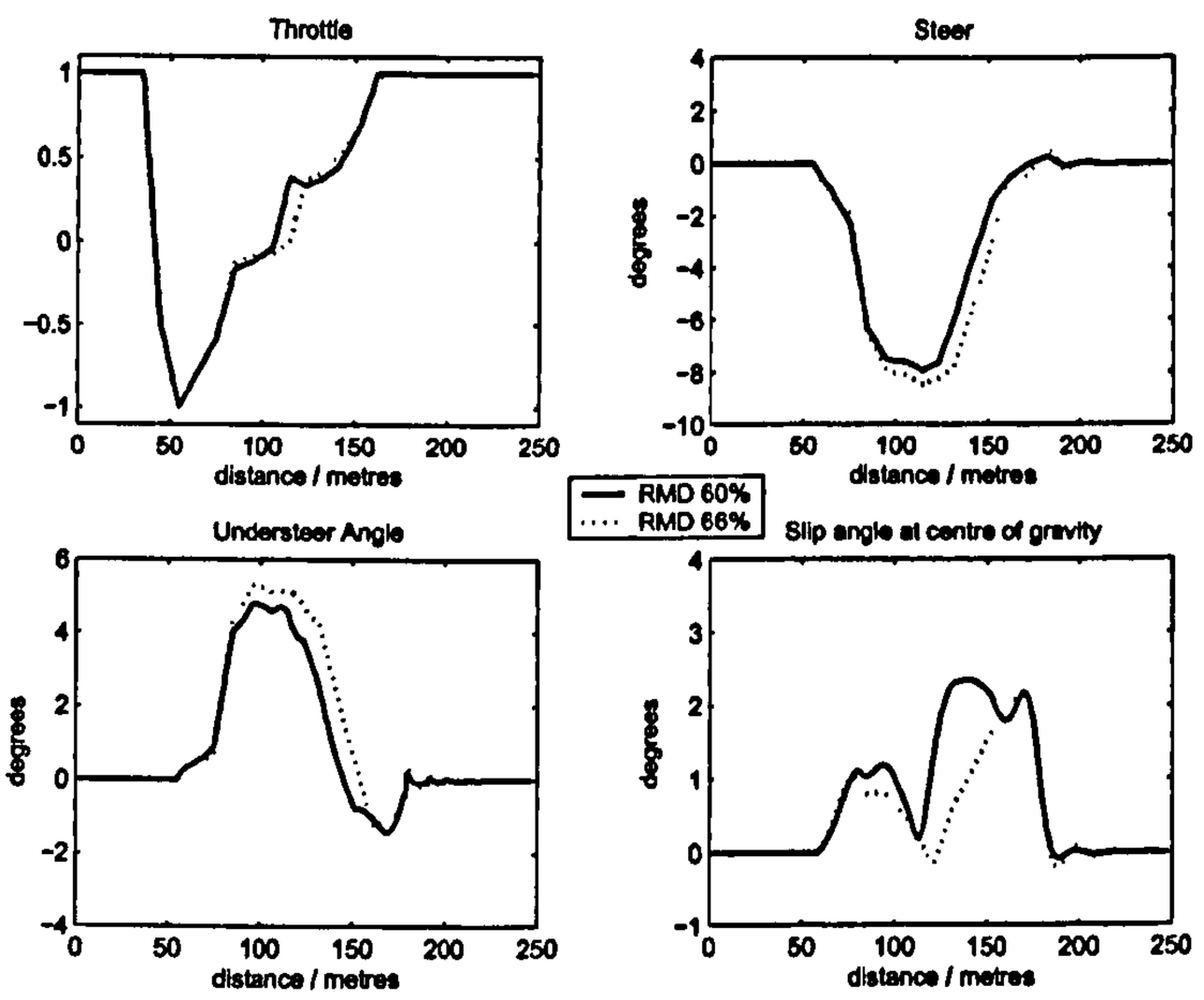


Figure 4.13: Roll Moment Distribution Effect on vehicle controls - Single 90° Corner

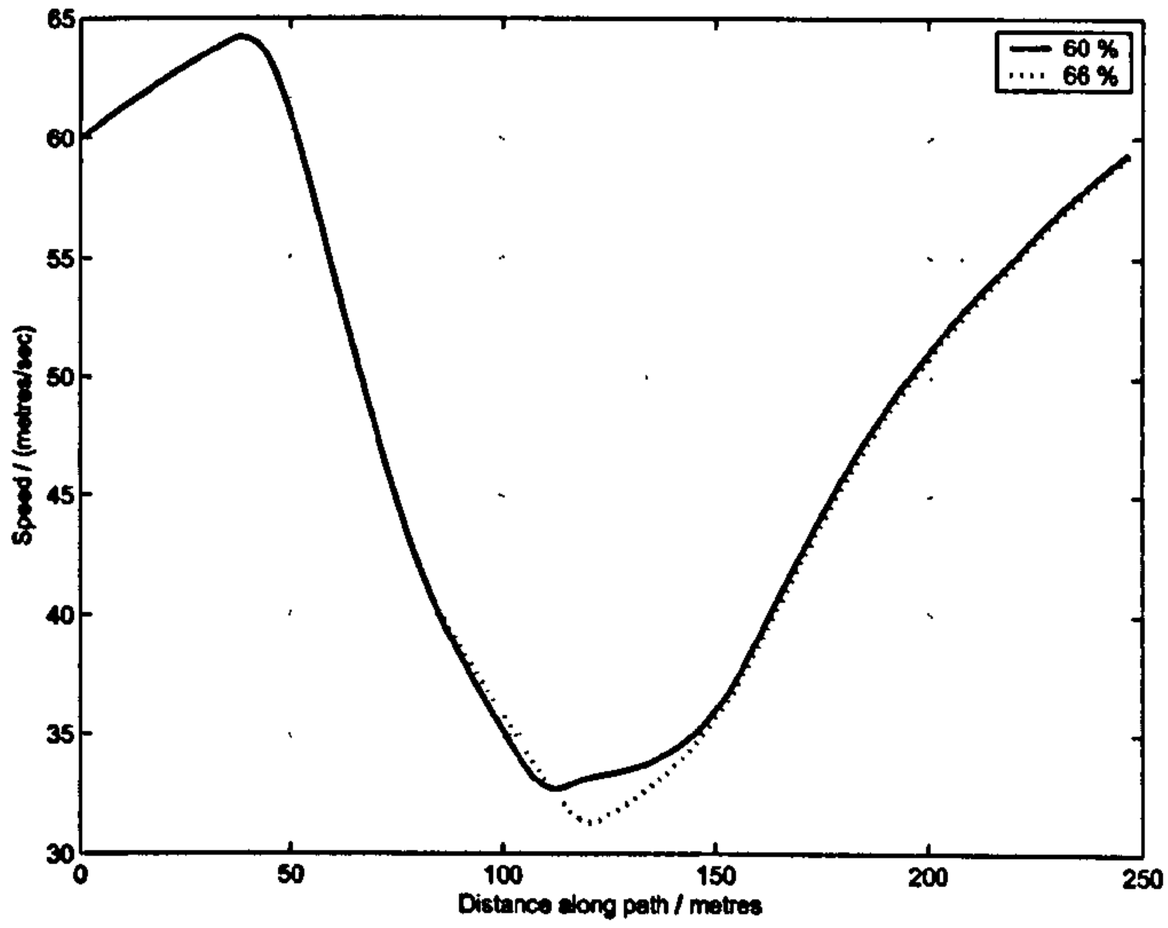


Figure 4.14: Effect of Roll Moment Distribution on Speed - Single 90° Corner

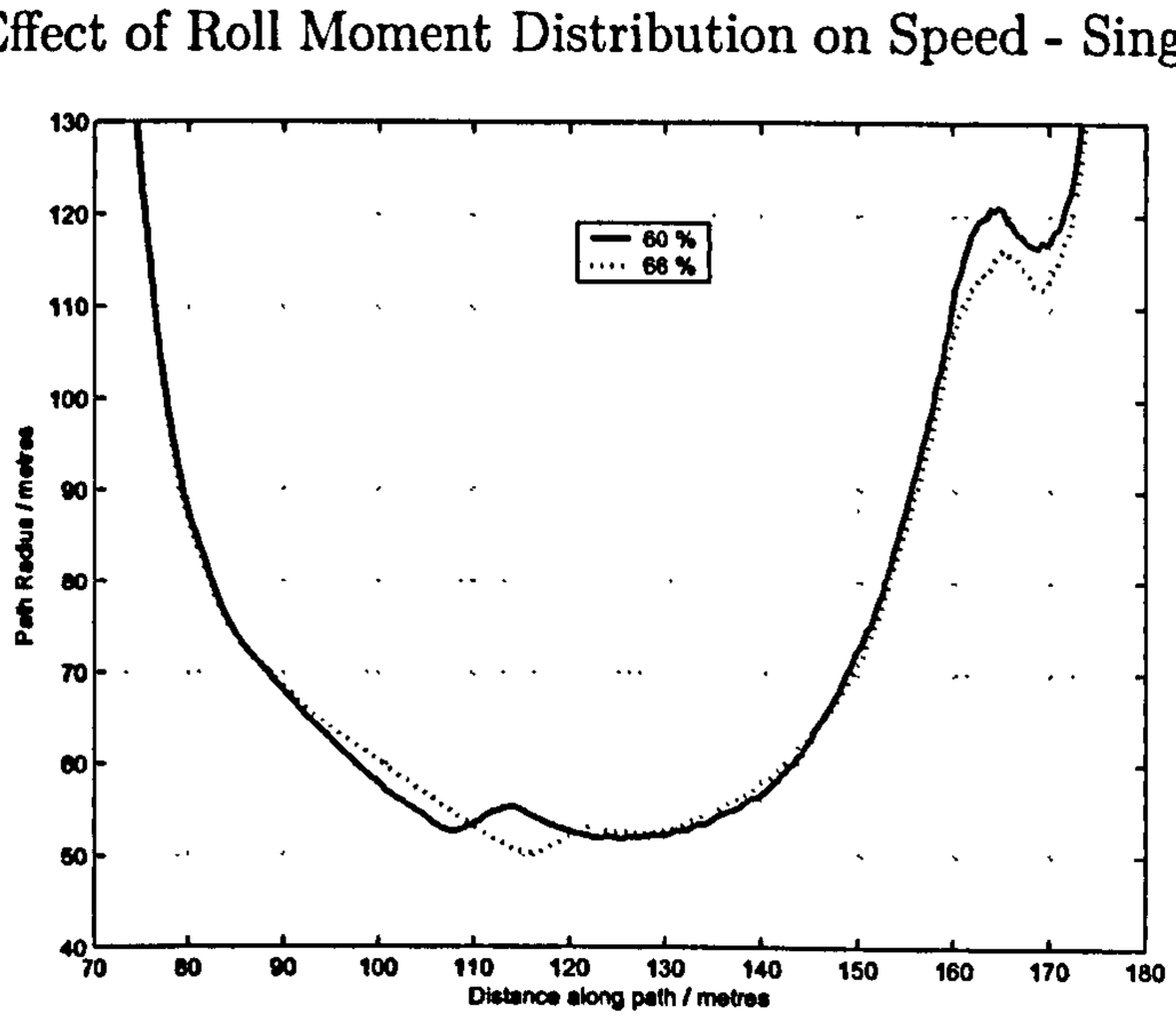


Figure 4.15: Effect of Roll Moment Distribution on Driven Line - Single 90° Corner

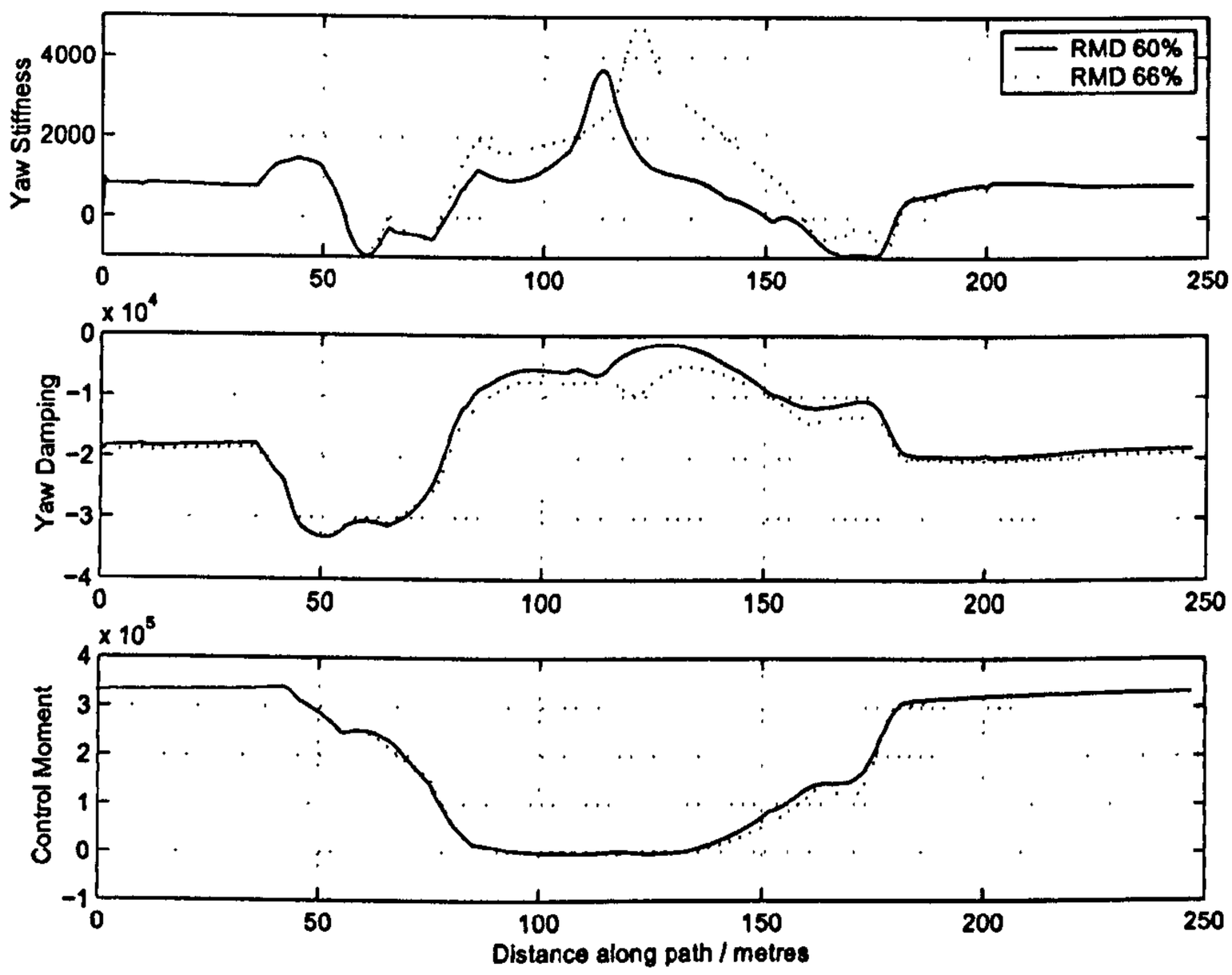


Figure 4.16: Effect of Roll Moment Distribution on Yaw moment Derivatives - Single 90° Corner

4.6 Longitudinal Centre of Gravity Location

Here we consider the effect of the longitudinal location of centre of gravity (weight distribution).

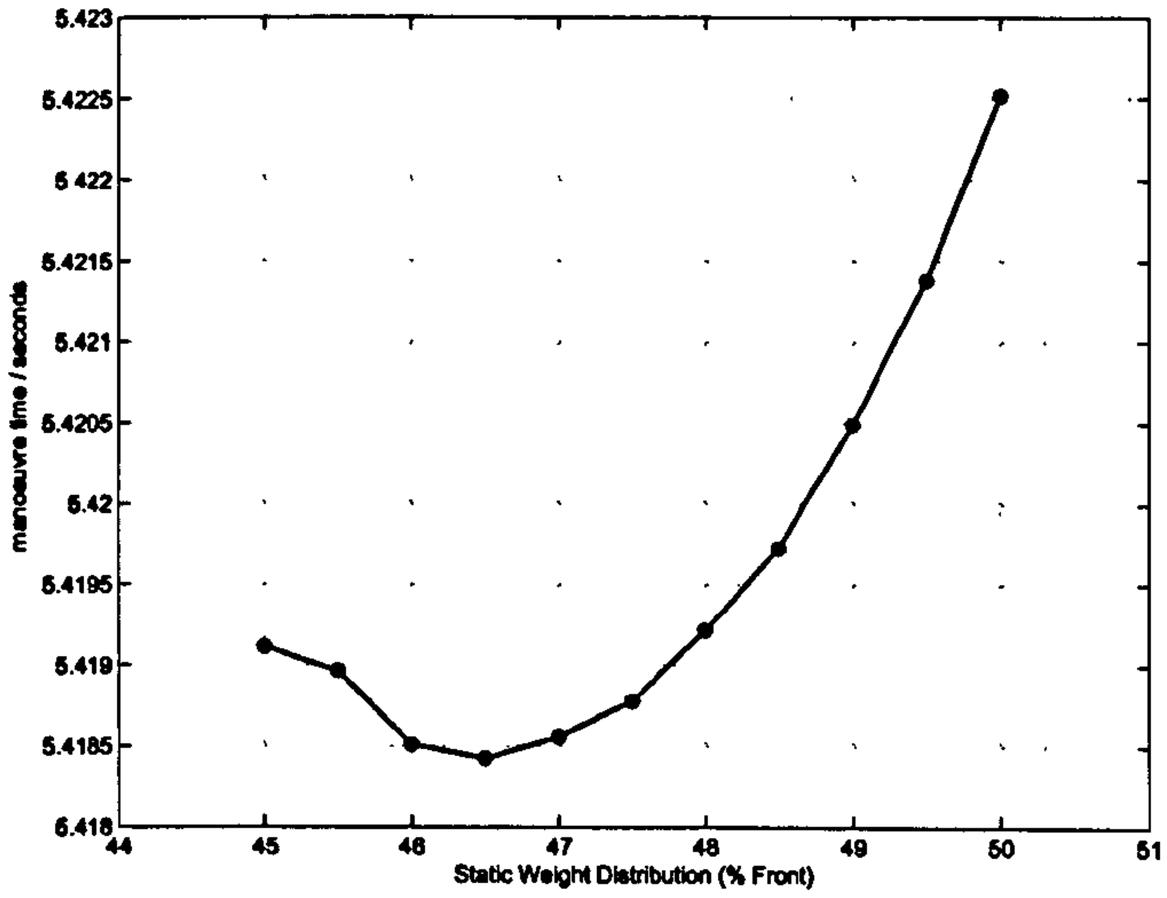


Figure 4.17: Effect of Static Weight Distribution on manoeuvre Time - Single 90° Corner

manoeuvre	Time Delta (ms)	Time Delta %	Time Delta %/%
90° corner	3.4	0.063	0.013

Table 4.4: Change in manoeuvre time for weight distribution 45% to 50% Front

4.6.1 Vehicle controls

Figure 4.18 shows small changes in throttle and steer across the two extremes of weight distribution considered. The forwards weight shows slightly more understeer. The biggest difference is seen in the body side slip plots showing that the vehicle attitude changes significantly between the two cases.

4.6.2 Driven Line and Speed Profile

Figures 4.20 and 4.19 show the driven lines and speed profiles for the 45% and 50% weight distributions. The forwards-weight case turns in later and has a lower minimum speed.

4.6.3 Vehicle Stability

Figure 4.21 shows the yaw stiffness for three weight distribution cases. A step forward in weight distribution appears to apply a constant offset to the yaw stiffness. In the 50%

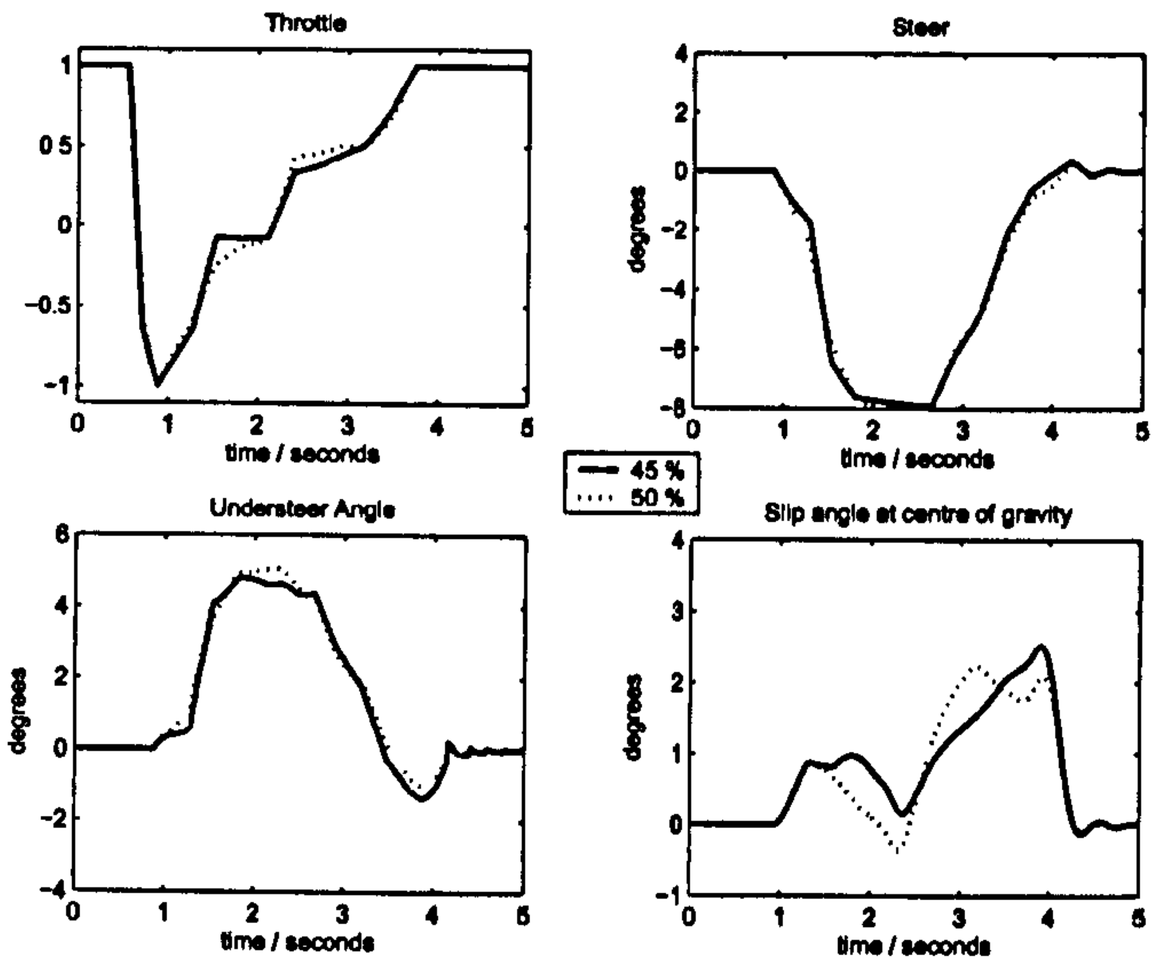


Figure 4.18: Comparison of forwards and rearwards weight - Single 90° Corner

case, the yaw stiffness remains positive for almost the entire manoeuvre. Figure 4.22 shows the yaw damping and control moment. As the weight moves forward there is increased yaw-damping in the trailing throttle period and decreased yaw damping as the throttle is applied.

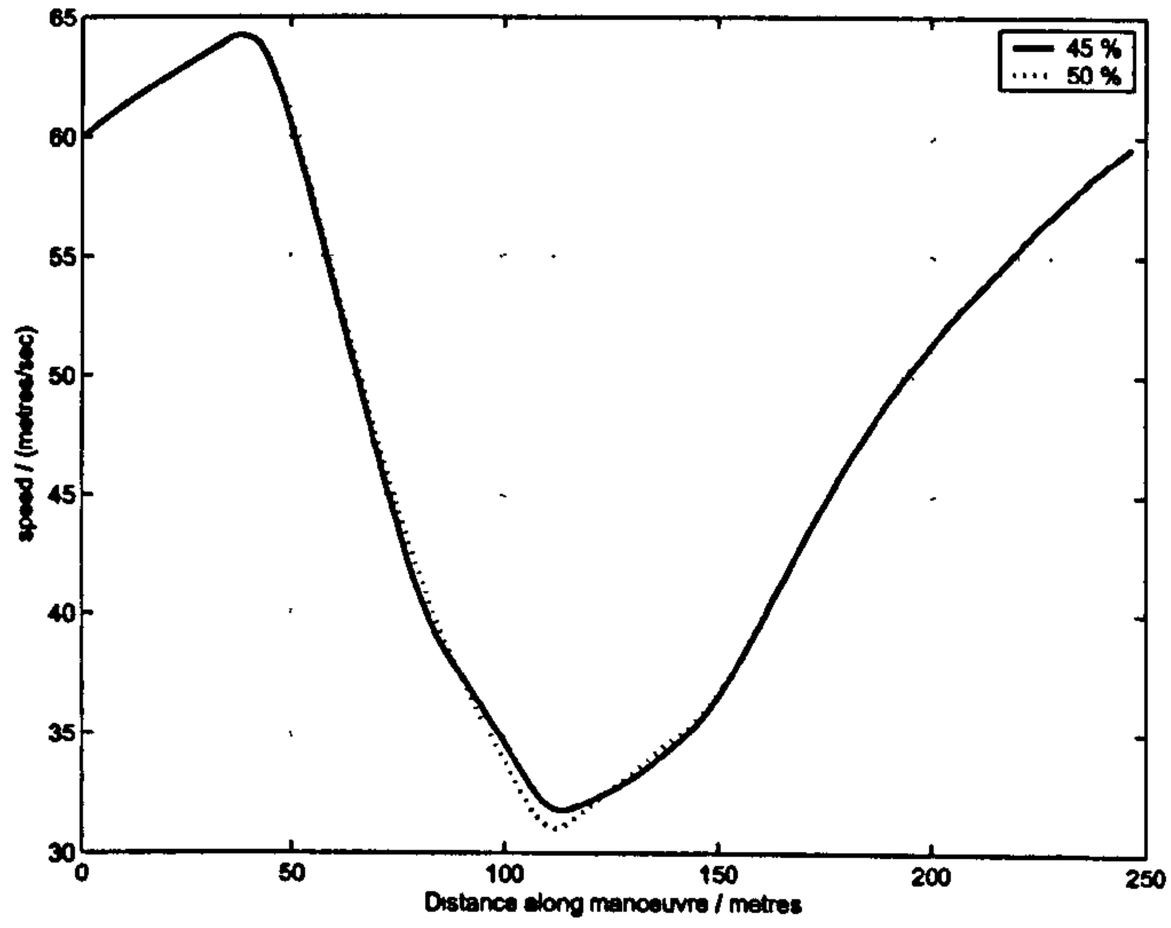


Figure 4.19: Longitudinal Weight distribution - Single 90° Corner

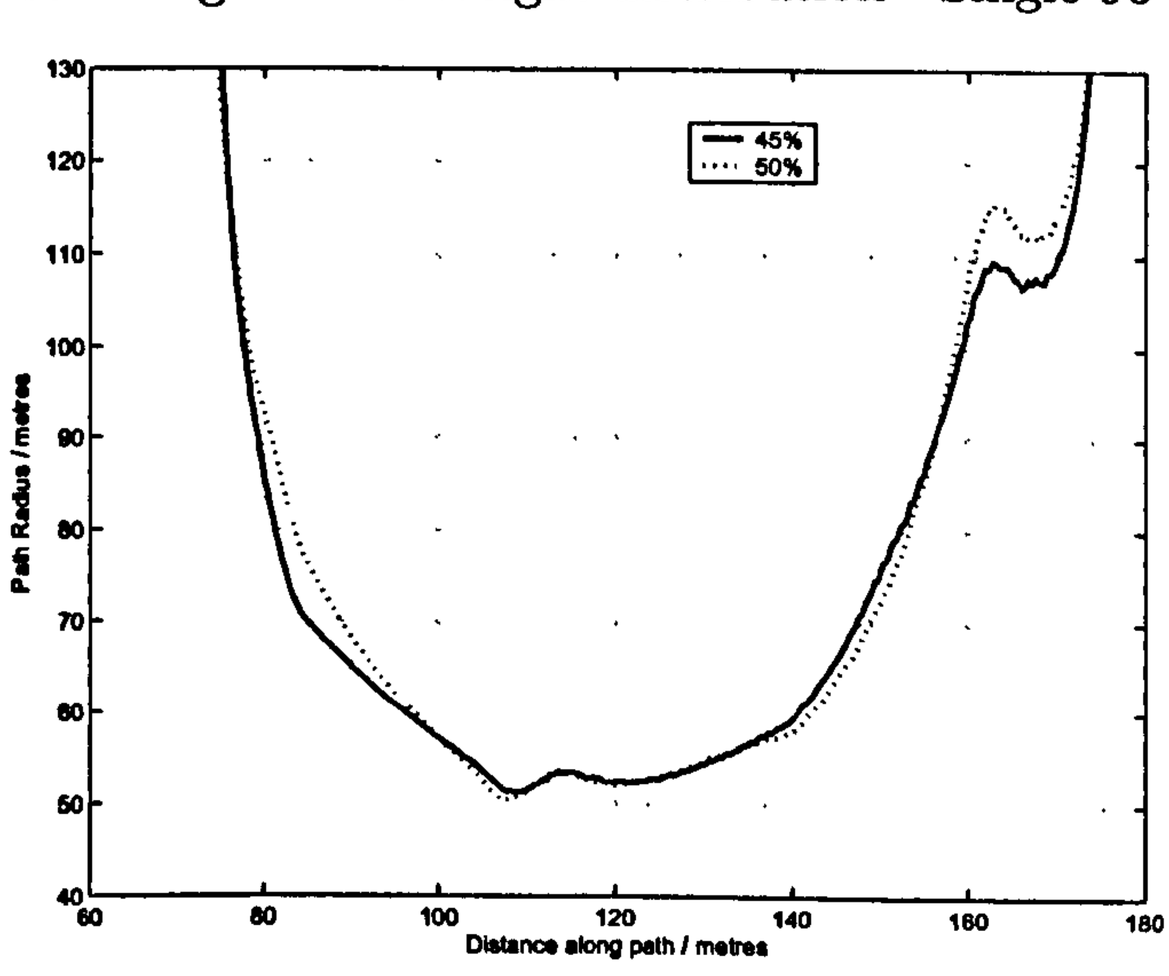


Figure 4.20: Longitudinal Weight distribution and Driven Path Radius - Single 90° Corner

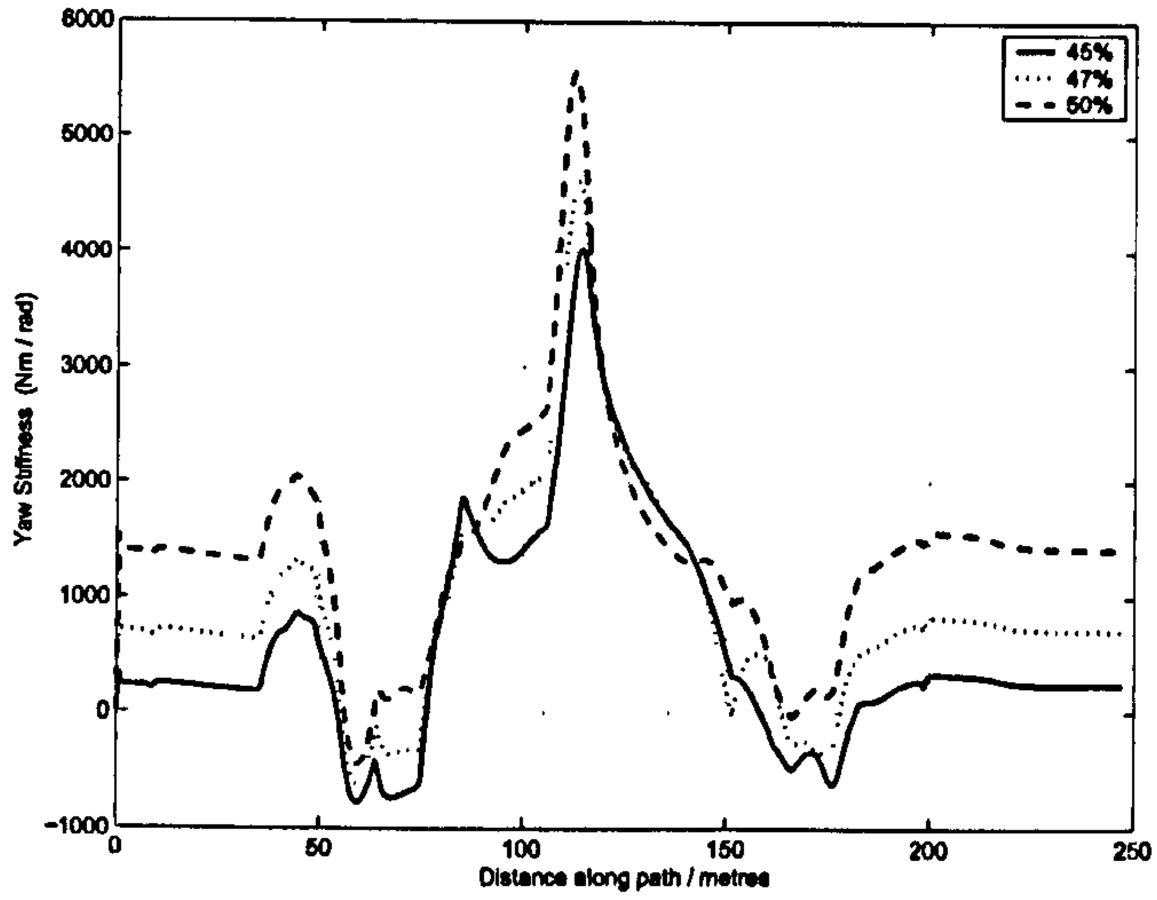


Figure 4.21: Longitudinal Weight distribution and Yaw Stiffness- Single 90° Corner

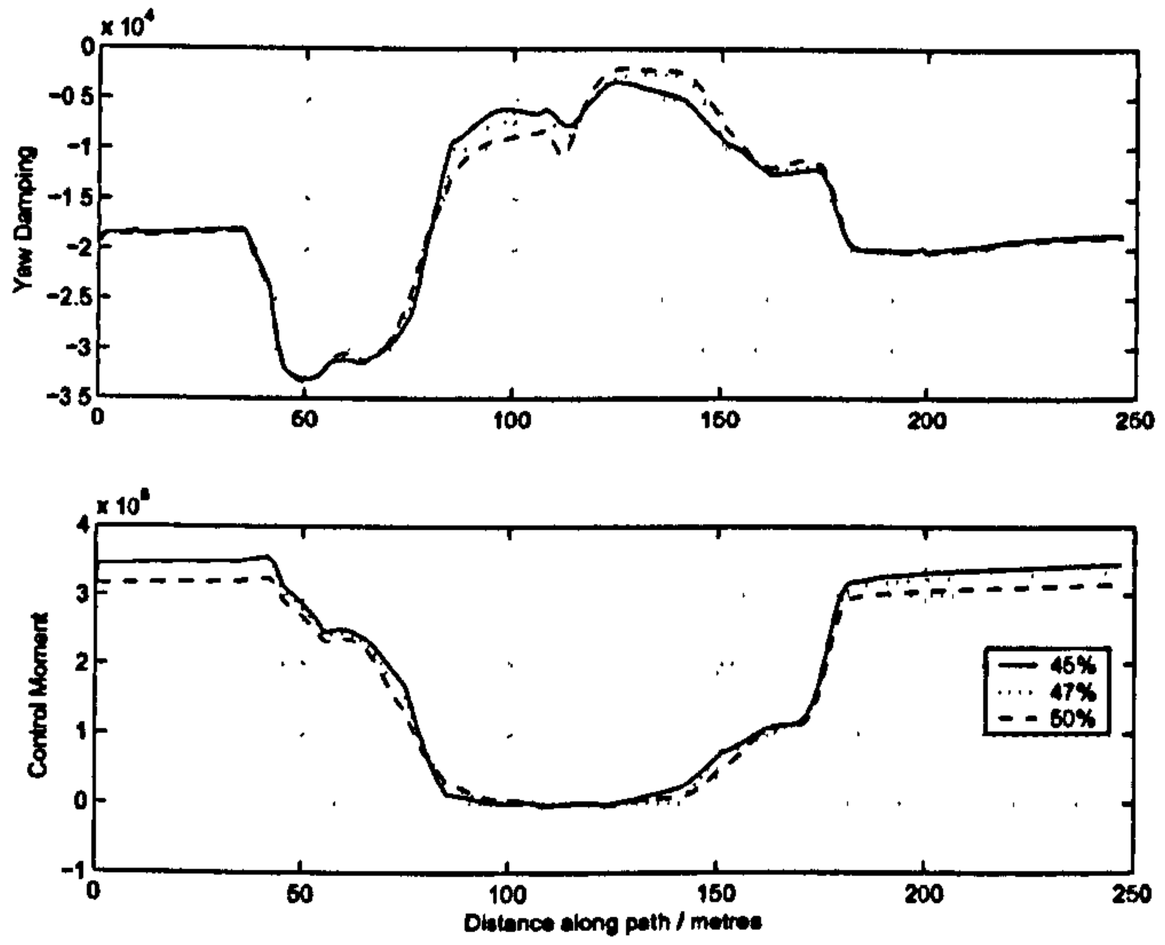


Figure 4.22: Longitudinal Weight distribution and Yaw Damping / Control Moment - Single 90° Corner

4.7 Differential

Here we consider the effect of the rear-axle differential set-up. The differential model is described in Appendix A. The torque transferred is a function of the engine input torque and the rear wheel speed difference. Four different differential set-ups are considered as described in Appendix A. The first two are the simple open and locked case. The second two are limited slip cases with LimSlipA offering more locking torque than LimSlipB. The vehicle set-up used is QSS/B with Tyre set ST/B. Figure 4.23 shows the manoeuvre time for each differential.

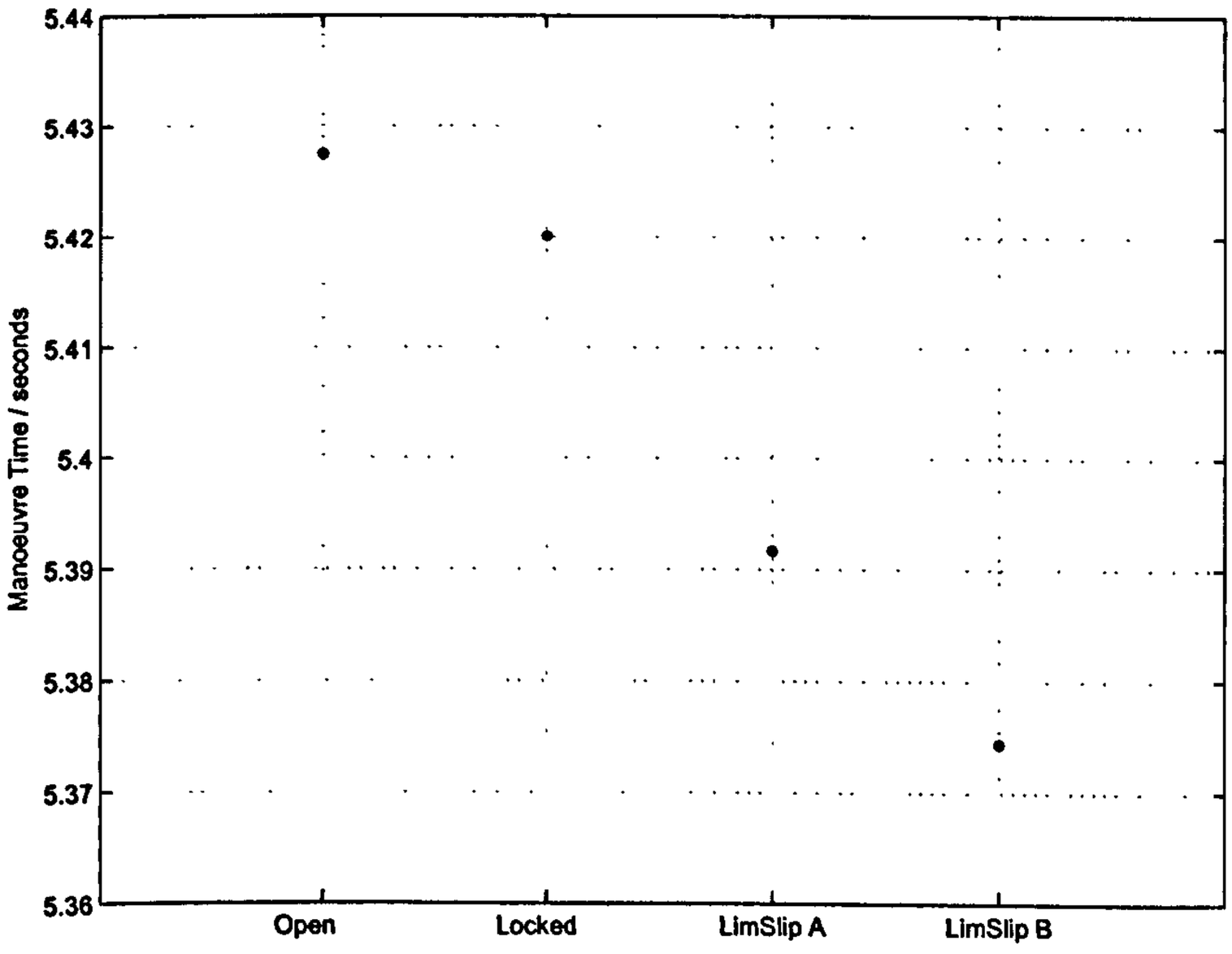


Figure 4.23: Impact of Differential Strategy on manoeuvre Time - 90 ° Corner

Strategy	Time Delta from Open Diff (ms)	(%)
Locked	7.4	0.14
LimSlip A	35.9	0.66
LimSlip B	53.1	0.98

Table 4.5: Differential Strategy effect on manoeuvre time

4.7.1 Vehicle controls

Figure 4.24 shows a comparison of the open differential and the limited slip B strategy. The open differential brakes to the potential of the unloaded rear wheel. As a result, the

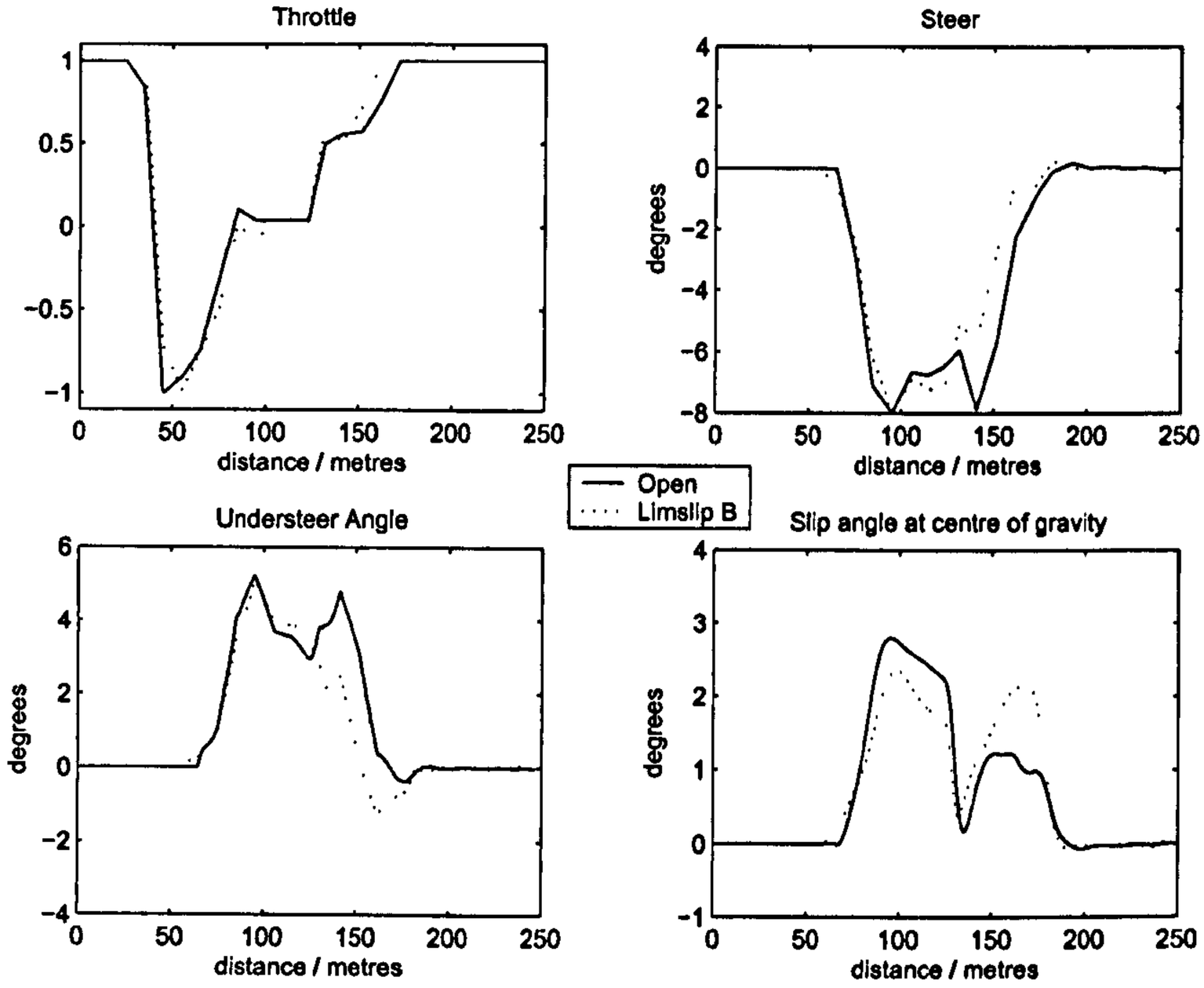


Figure 4.24: Effect of Differential set-up on Vehicle Controls - 90 ° Corner

open differential solution must brake earlier. The limited slip solution exits the corner with a more oversteer balance.

4.7.2 Driven Line and Speed Profile

Figure 4.25 shows the path radius of the optimized driven line for three differential strategies. A significant difference can be seen between the driven lines. The open differential achieves a minimum path radius approximately 5m less than that of the locked. The limited slip differential follows a similar path to the locked differential on turn entry. As the Apex is approached it assumes a path radius closer to that of the open differential. In then rejoins the locked differential path as the throttle is applied and the car exits the corner.

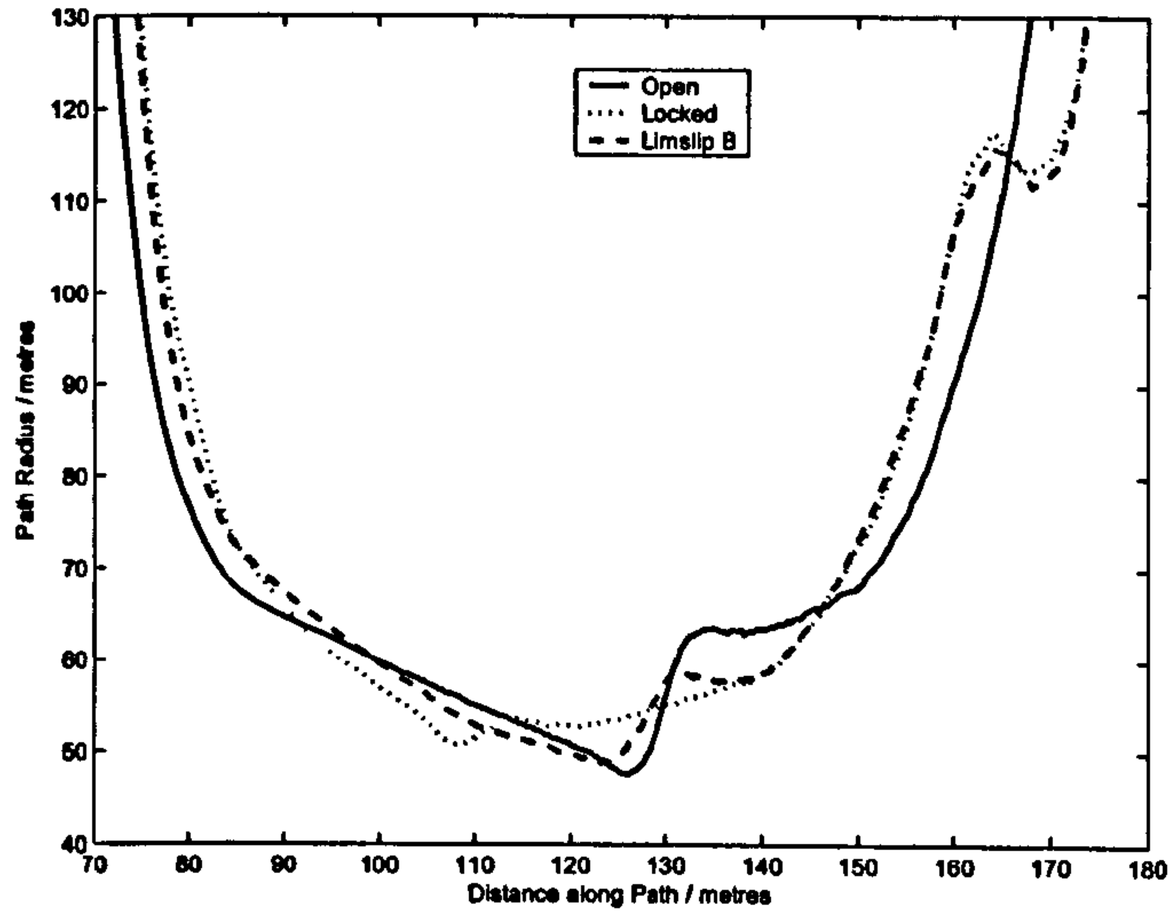


Figure 4.25: Effect of Differential set-up on Driven Line - 90 ° Corner

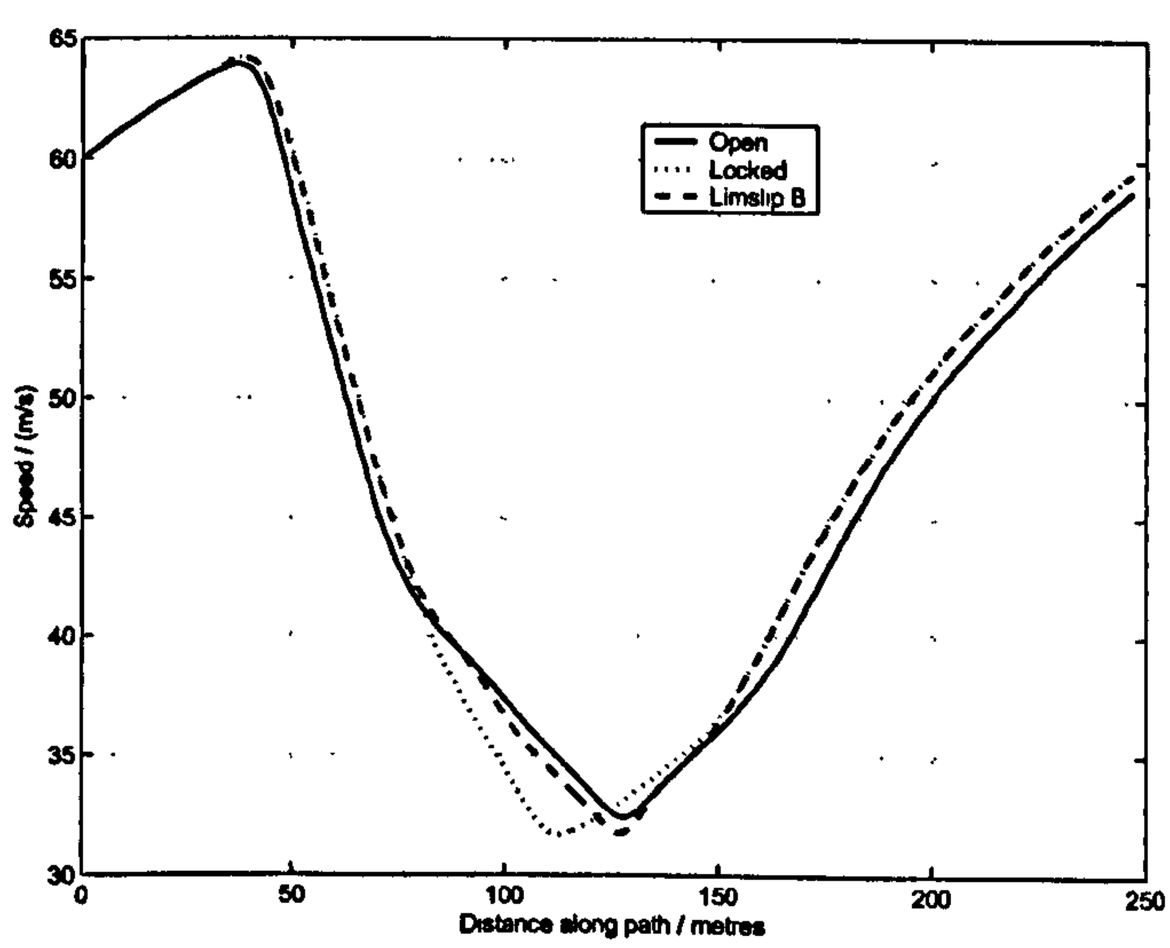


Figure 4.26: Effect of Differential set-up on Speed Profile - 90 ° Corner

4.8 Rear Grip Scale Factor

Here we consider the effect of a reduction in rear tyre grip. This is achieved by applying a scaling factor to the longitudinal and lateral forces generated by the rear tyres as explained in Appendix B. A reduction in rear grip is generally associated with a reduction in vehicle stability. It is useful when analysing a vehicle set-up to consider the impact of small changes to front and rear grip. During the course of a racing event changes to the conditions of the track and its temperature will change the available tyre grip. This is as a result of complex thermodynamic processes in the tyre that typically result in a different change in grip at the front and rear axles hence a change in the limit balance of the car. It is common for the same vehicle set-up to produce a balanced car on one day and an unstable oversteer car on the next. By studying the effect of grip changes on a set-up it is possible to test the robustness of the set-up to these environmental factors.

4.8.1 Vehicle Controls

Figure 4.27 shows vehicle controls, understeer and body side-slip angles for the grip scale cases of 1.0 and 0.95. The low rear grip controls show earlier braking and a sharp initial turn-in on the steering. This sharp turn-in results in a higher initial understeer angle which then settles to less understeer through the Apex of the corner.

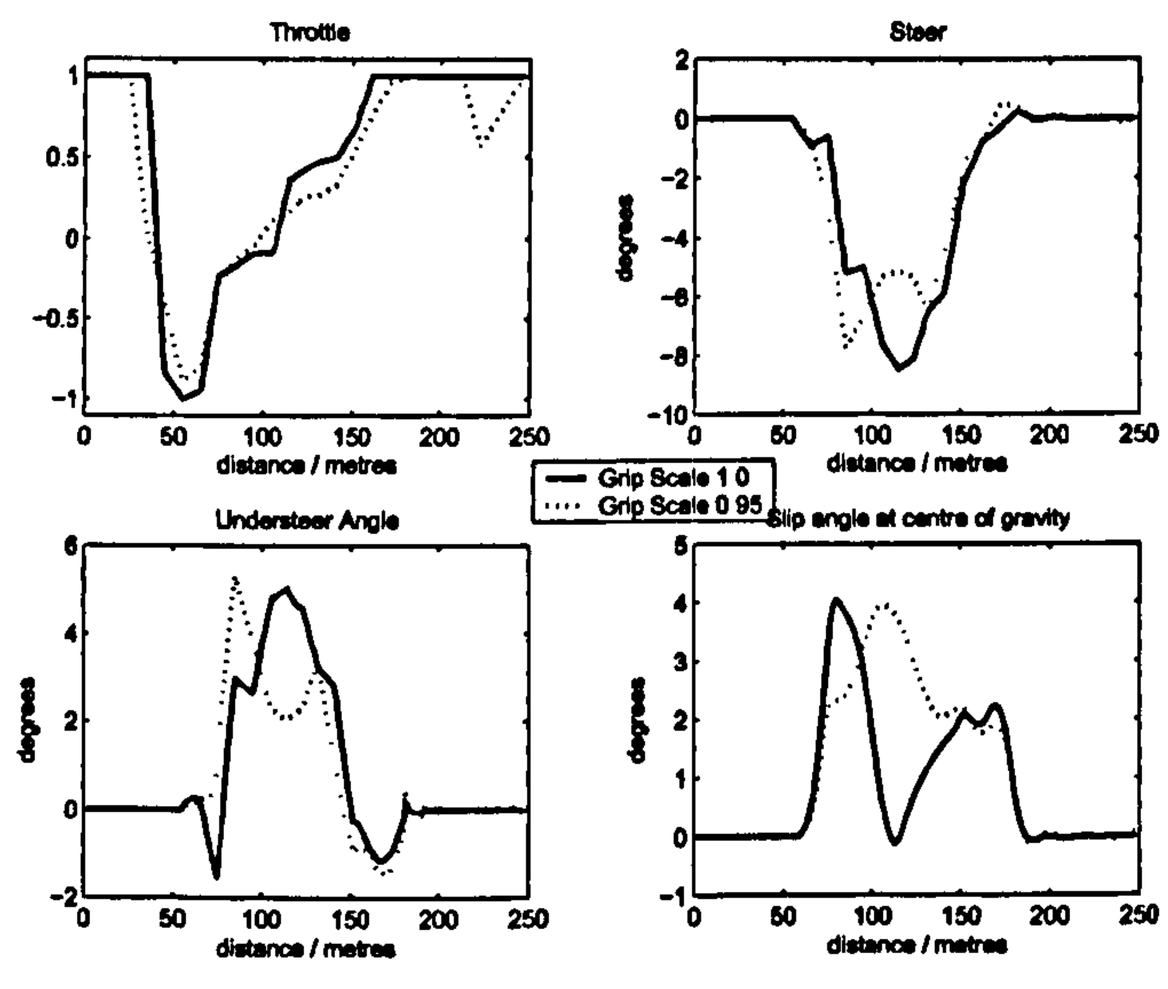


Figure 4.27: Grip Scale - Single 90° Corner

4.8.2 Driven line and speed profile

Figures 4.28 and 4.29 show the drive line and speed plots for the three cases. The 0.975 and 0.95 cases both turn into the corner earlier than the 1.0 case. Perhaps surprisingly, the lower grip cases choose a higher minimum speed at the expense of a decrease in exit speed.

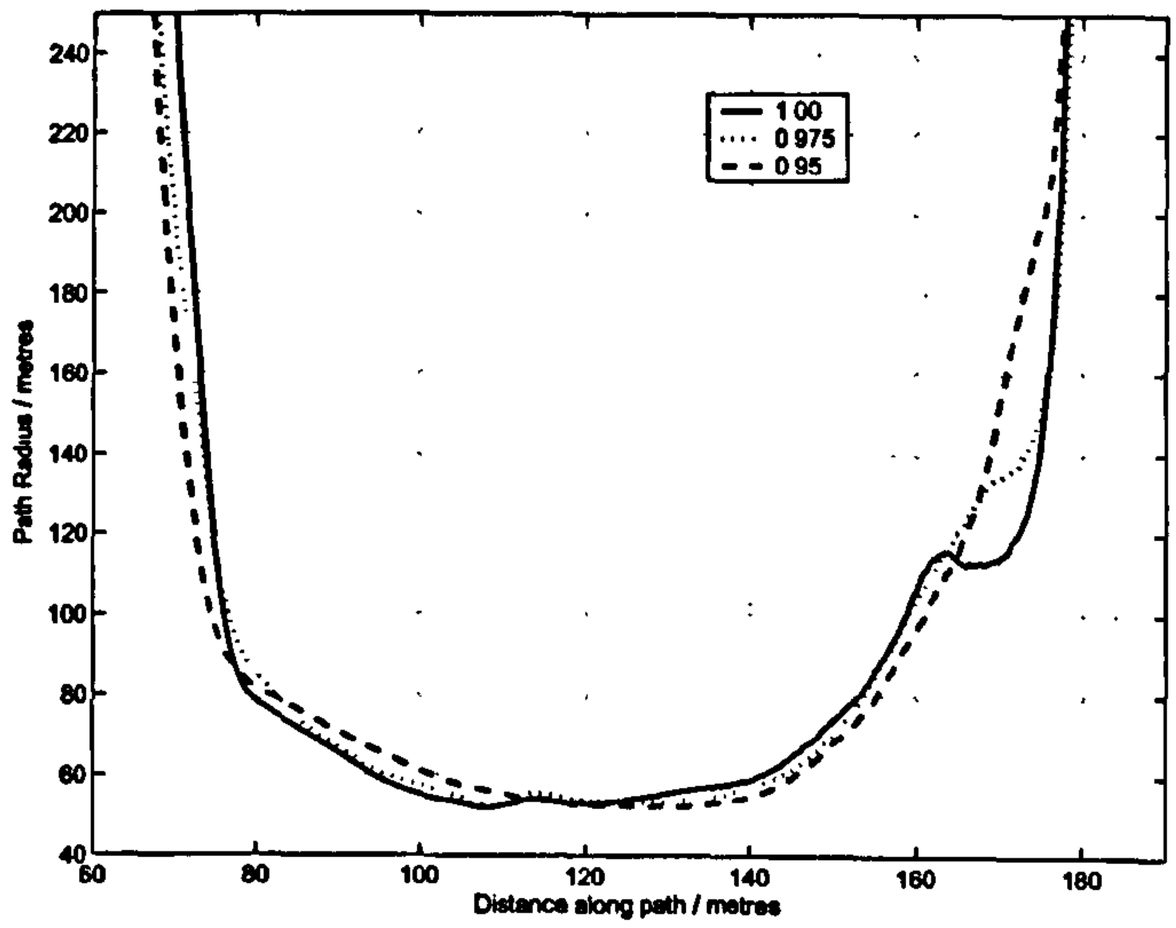


Figure 4.28: Effect of rear tyre grip on Driven Line - 90 ° Corner

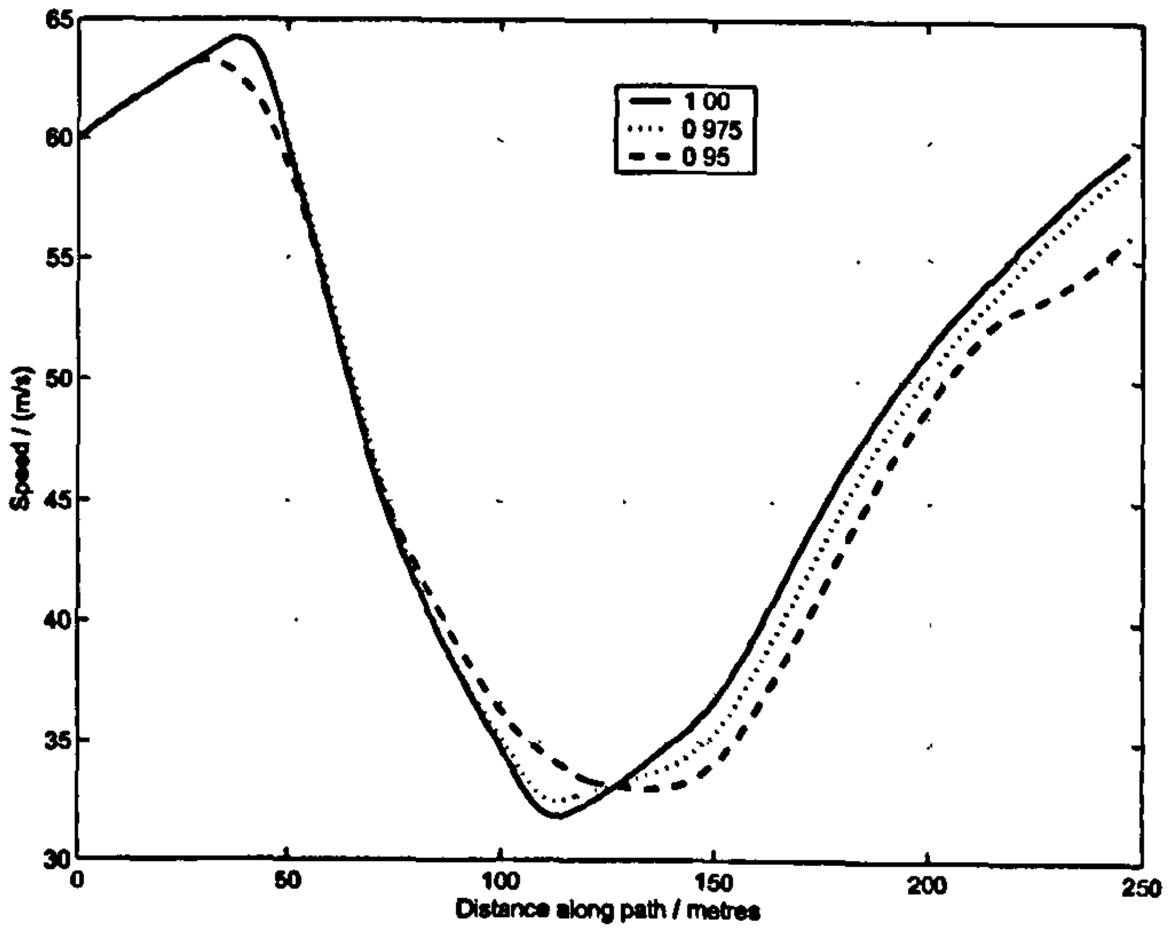


Figure 4.29: Effect of rear tyre grip factor on Speed Profile - 90 ° Corner

4.8.3 Vehicle Stability

Figure 4.30 shows the yaw stiffness for rear tyre grip scale factors of 1.0, 0.975 and 0.95. As expected, the yaw stiffness is reduced following a reduction in rear tyre grip. The yaw stiffness change corresponding to grip scale factors of 1.0 and 0.975 (a 2.5% reduction in grip) is approximately equivalent to a rearwards shift in weight distribution of 1.5-2%. At the Apex of the corner, it is equivalent to a rearwards movement of roll moment distribution of approximately 5%. In the 0.95 case the yaw stiffness is negative or close to negative indicating that the vehicle is open loop unstable through most of the manoeuvre.

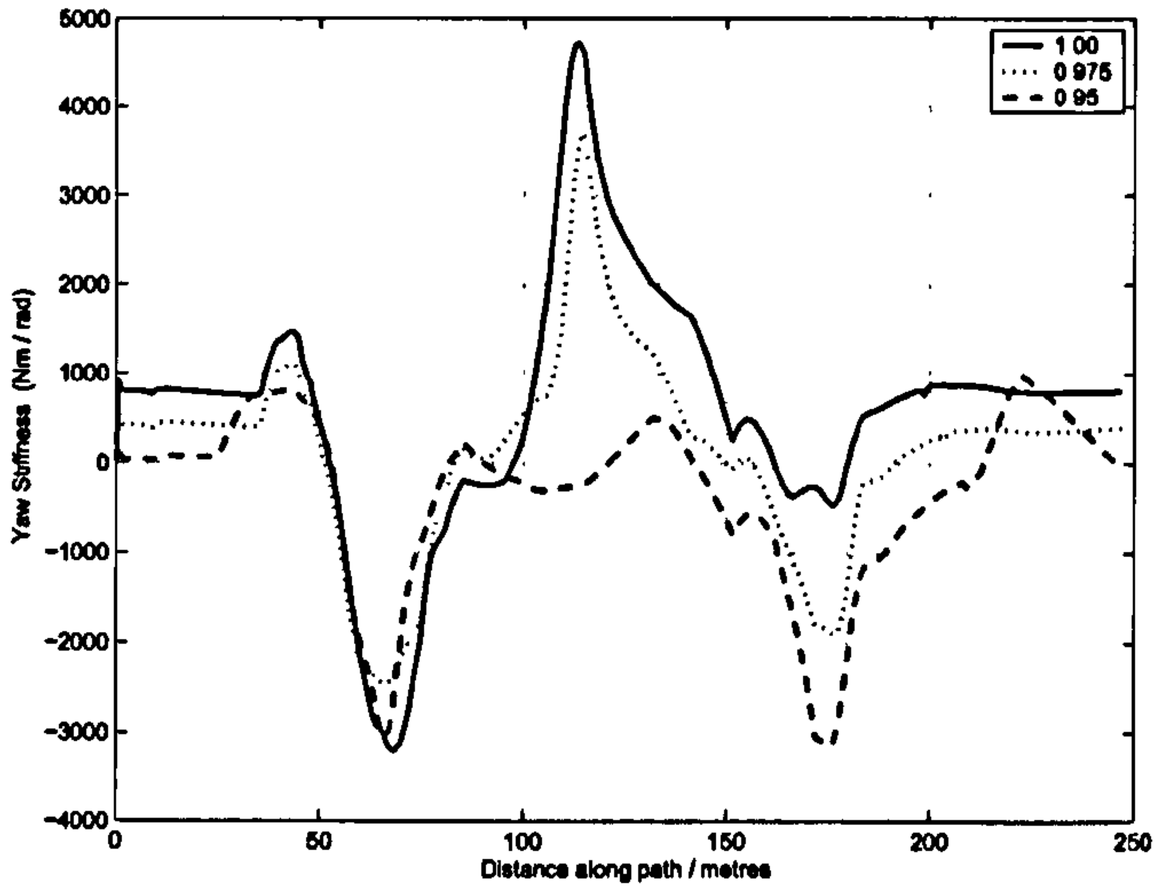
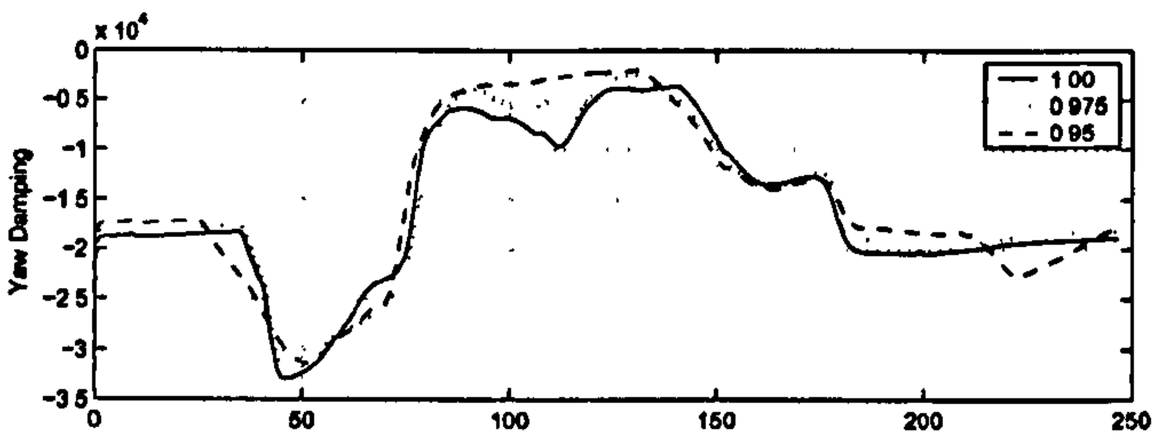


Figure 4.30: Grip Scale and Yaw Stiffness- Single 90° Corner



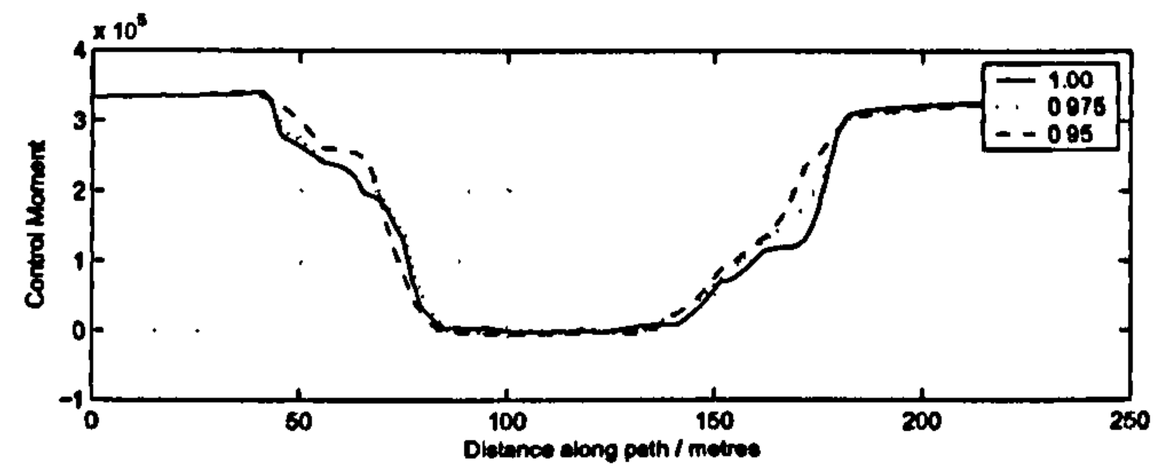


Figure 4.31: Grip Scale and Yaw Damping / Control Moment - Single 90° Corner

4.9 Influence of a Straight Following a Corner

In this section we take a brief look forward to Chapter 6 by considering the change in approach to a corner when it is followed by a straight. We find optimal solutions for the righthander with manoeuvre lengths from 200m to 350m in increments of 10m. From around 200m onwards, the final controls are approximately full throttle and zero steer. The final rotation of the car is such that if the trajectory is extended with zero steer values, the car continues along the straight that follows the corner without violating the displacement contraints. We extend all of the resulting trajectories to length 350m by appending zero steer and full throttle control values. The resulting manoeuvre times give some insight in to far ahead along the track the driver must plan in order to minimise manoeuvre time. Figure 4.32 shows a plot of the resulting 350m manoeuvre times and final speeds against

the original optimized manoeuvre length in the range 200-340m. As could be expected, the manoeuvre time reduces as more of the exit straight is 'seen' during the optimization process. For manoeuvre lengths in the 250-270m range two solutions exist and this appears to be a 'cross-over' region where the optimizer can see enough of the straight to see the benefit of higher exit speed from the corner. Figure 4.33 gives the driven line and speed plots against distance along the manoeuvre for the 220m and 240m cases. The 340m case brakes earlier and has the peak in path curvature approximately 10m earlier. This topic is covered in more depth in Chapter 6.

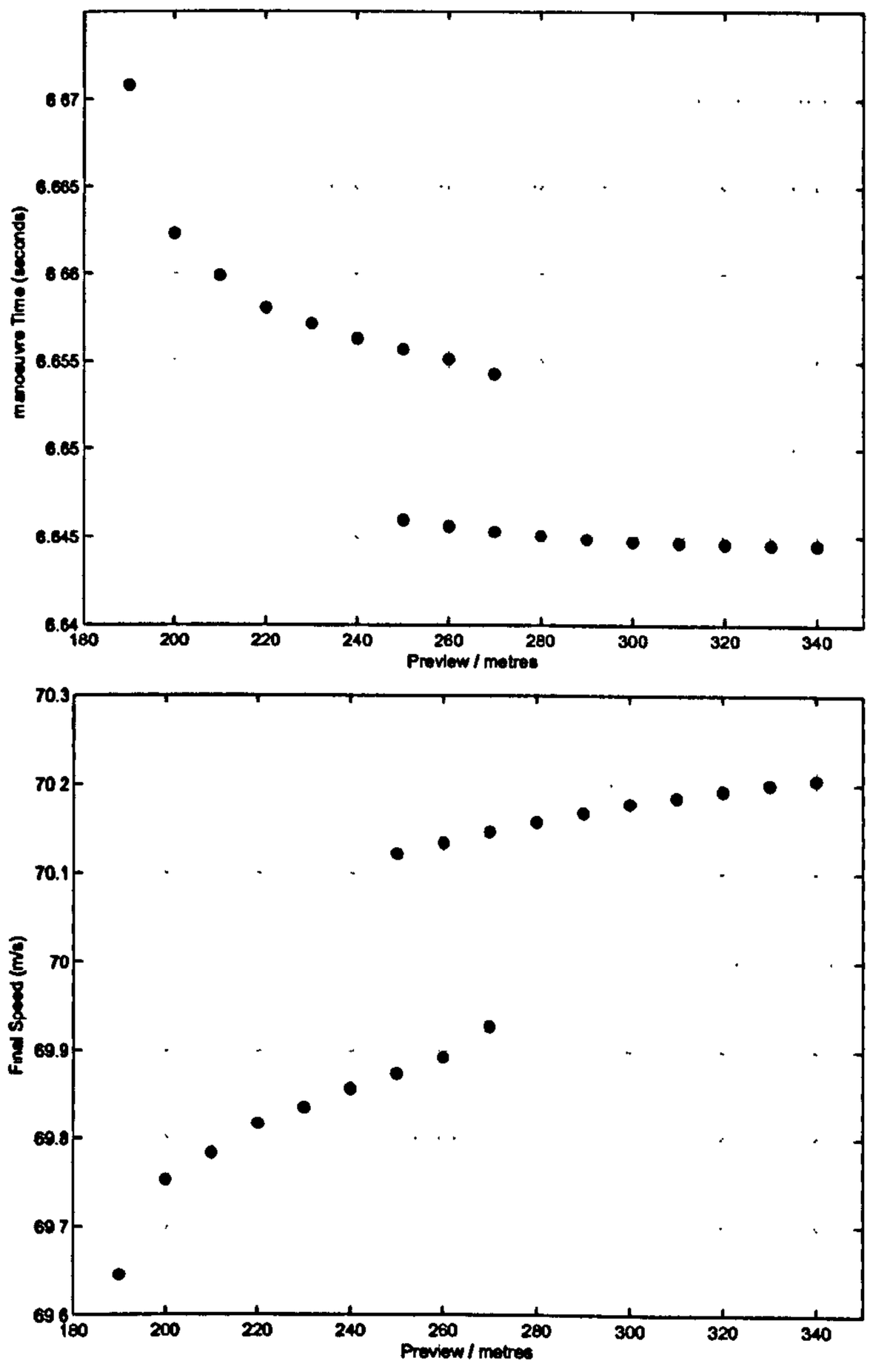


Figure 4.32: Influence of preview on manoeuvre time and Final Speed

4.9.1 A comparison of Minimum time and Maximum exit speed strategies

We have seen that there is a strong link between minimum manoeuvre time and manoeuvre final speed. Here we consider changing the objective to that of maximising the final speed.

	Minimum Time	Maximum Speed
Time	6.644509	7.18615462
Speed	70.205503	70.85347808

Table 4.6: Comparison of Minimum Time and Maximum exit speed

We modify the MTM method simply by specifying the value of the objective function as minus the vehicle speed on crossing the final waypoint in the manoeuvre. All other aspects of the algorithm remain unchanged. Figure 4.34 shows an overlay of the driven lines for the minimum manoeuvre time and maximum exit speed cases. Table 4.6 shows the manoeuvre times and final speeds for each case. The final exit speed generated by the maximise-exit-speed approach is slightly higher but the resulting manoeuvre time is significantly higher. The maximise-exit-speed has not been studied in any depth by the author but this simple example does show that the blind pursuit of exit speed as an objective does not result in a good approach to the minimum-time-manoeuvring problem.

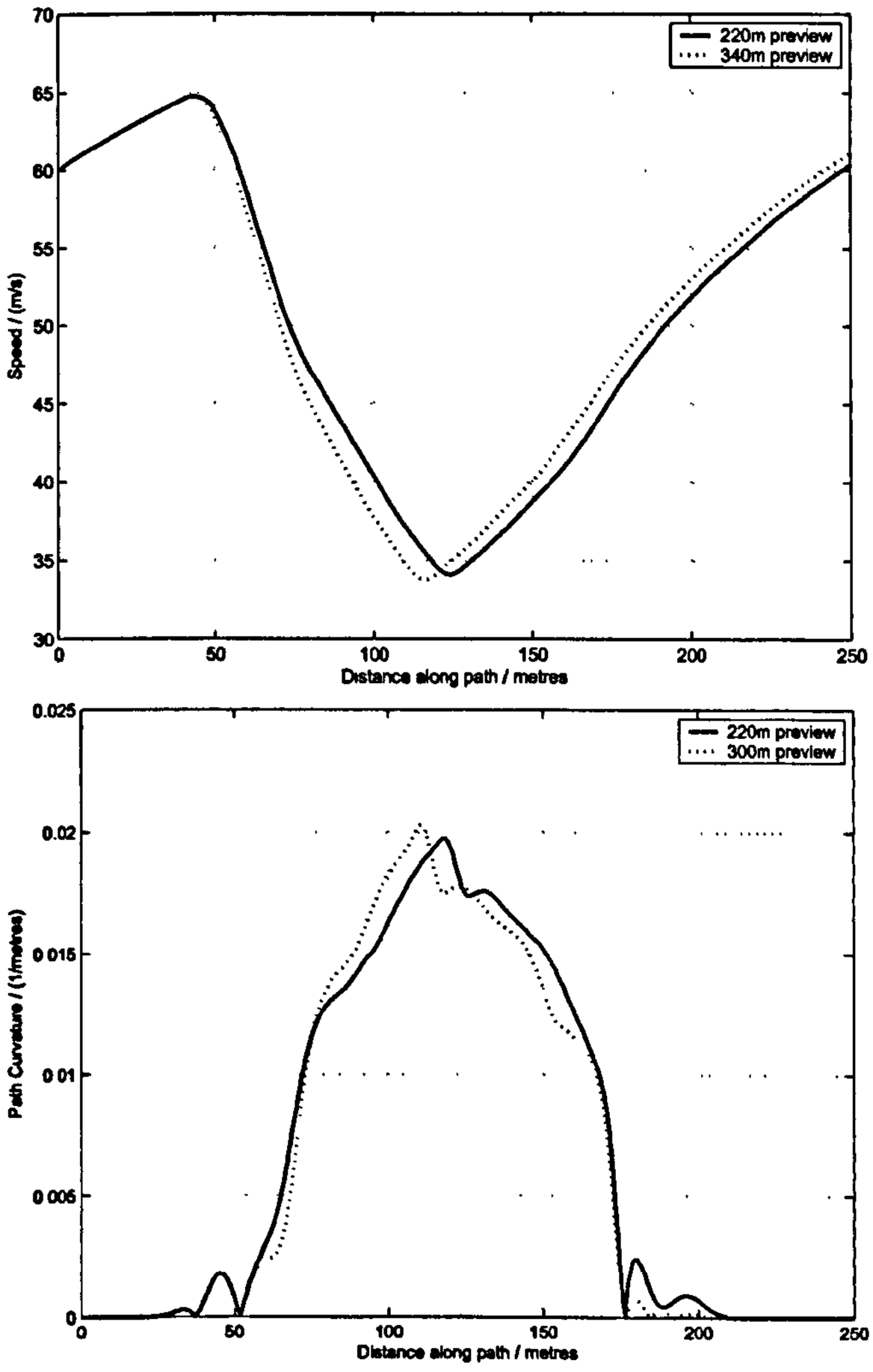


Figure 4.33: Comparison of Speed and Path Curvature for 220m and 340m solutions

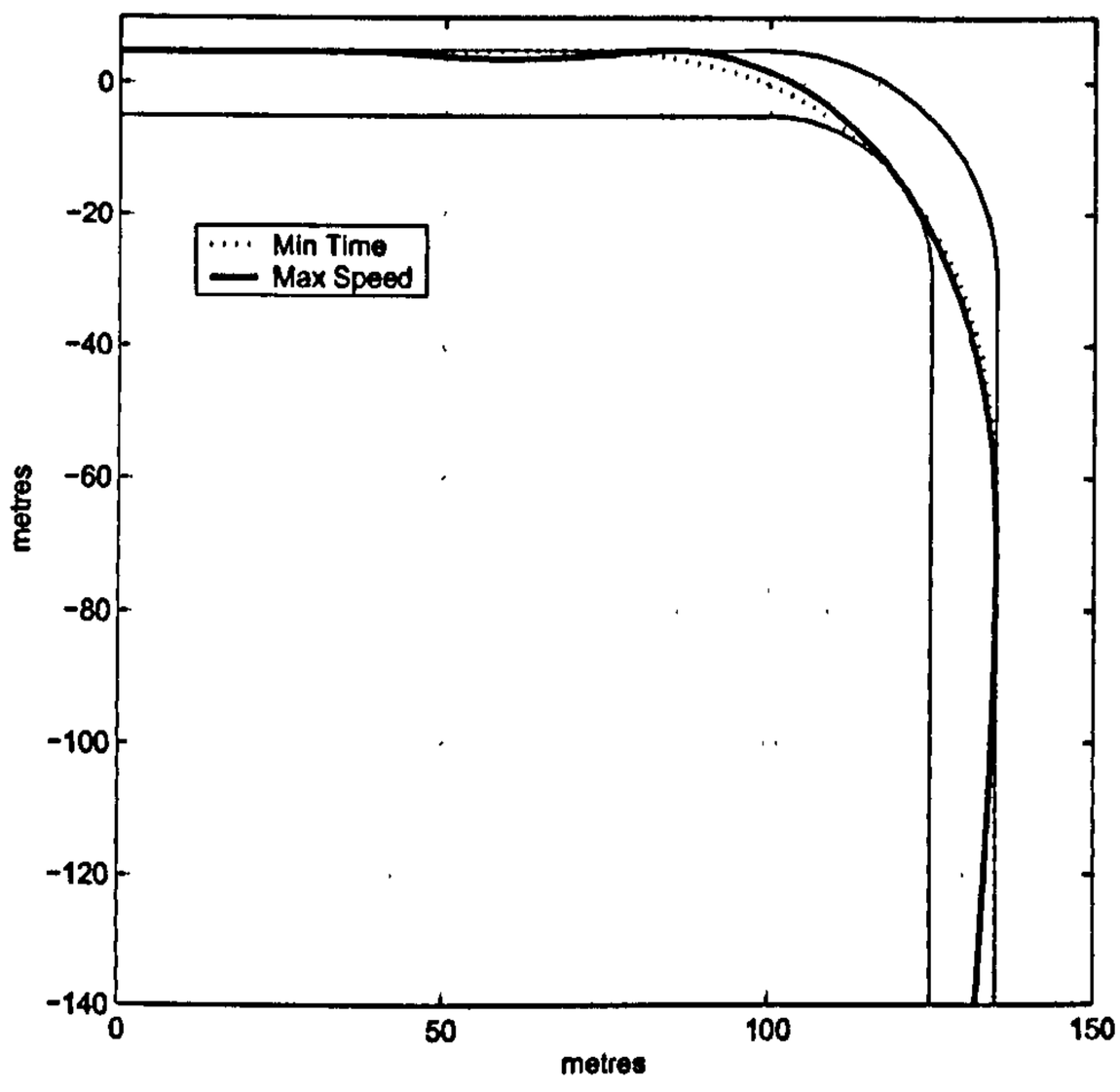


Figure 4.34: Comparison of Driven line for Minimum time and Maximum exit speed

Chapter 5

Comparison of Transient and Steady State Methods

If everything seems under control, you're just not going fast enough

Mario Andretti

5.1 Summary

In this chapter, we compare the results of the MTM method with those produced by a Quasi-Steady-State manoeuvre time simulator as described in Appendix D. We examine the effect of a change of driven line on the results produced by the QSS method.

5.2 Comparison of Transient MTM and QSS Solutions

In this section, we compare MTM and QSS solutions for the single $90°$ corner. The MTM result is obtained using vehicle set-up QSS/A and tyre set A. We use a 5m control spacing with the steer values held at zero until the 55 metre line. The QSS result is obtained using the path curvature data produced by the MTM method sampled at intervals of 1 metre. The 'apex' for the QSS solution is taken as the point of maximum curvature. The apex solution is obtained assuming zero longitudinal acceleration. The QSS solution is computed using very small tolerances on residual moments (1^{-10}) and can hence be considered (at least in purely numerical terms) very accurate. Figures 5.1 and 5.2 show a comparative speed trace and timeline. Figures 5.4 and 5.5 show comparisons of vehicle controls. The small speed difference in the first 40-50m of the manoeuvre is due to the vehicle initial state values as the start of the manoeuvre for the MTM solution. In the MTM solution, the procedure is presented with an initial vehicle state including rear tyre slip ratios that are

not set to optimal values. Hence the rear wheels must be accelerated up to speed before optimal acceleration can begin. The QSS solution gains time by braking later and reaching maximum brake force earlier. It then loses time towards the middle of the corner as it reaches a lower minimum speed. This is possibly related to the chosen apex position and the zero longitudinal acceleration assumption at the apex. It then gains time throughout corner exit. The QSS predicted manoeuvre time is approximately 38 ms faster or 0.77%. This can in part be explained by the increased controller bandwidth used. The QSS method has a control discretization interval of 1m and the MTM method used here 5m.

The results produced by the two methods are in fairly close agreement with the exception of the steering plot. The QSS procedure produces a very erratic steer trace. The car enters the corner with an oversteer balance and with opposite lock applied. With the QSS method, the vehicle balance can change on a point to point basis as there is no coupling between the vehicle state at the solution points other than the constraint that the longitudinal acceleration be sufficient to match the speed at the neighbouring point. As a result of this, the vehicle balance switches rapidly from point to point through the manoeuvre. Since the driven line is produced by the transient MTM solution, we can have some confidence that it is optimal.

The path curvature data is shown in figure 5.6. It can be seen that following the Apex, the curvature data is not in fact monotonic. As positive engine torque is initialy applied, weight is transferred to the rear of the car. For a short period, the car develops understeer and the path curvature is decreased. As engine torque then increases, the longitudinal utilisation of the rear tyres increases, their lateral capability decreases, the vehicle balance moves back towards oversteer and the path curvature increases. The racing line is studied in more detail in chapters 3 and 6.

In the QSS solution there are no unbalanced moments at each solution point. The vehicle has no yaw acceleration throughout the manoeuvre. Figure 5.3 shows the yaw rate for the MTM solution. It is interesting to note that the steer trace produced by the QSS solution is most different from the MTM solution when there is significant yaw acceleration.

Implicit in the QSS approach is the assumption that if a vehicle has a steady state solution at a given speed u_a in a given turn radius r_a at a given longitudinal acceleration, then it will also have a steady state solution at some speed $u > u_a$ for any given turn radius $r > r_a$ given the same longitudinal acceleration. If this assumption is true, a QSS solution can be found for any corner with monotonically increasing curvature up to the apex and monitonically decreasing curvature thereafter. This assumption is not always true in the case of a vehicle model equipped with non-linear tyres and a differential. The g-g diagrams shown in figure D.1 show the full range of maximum lateral and longitudinal vehicle accelerations at a number of fixed speeds. The plot for $40ms^{-1}$ shows that at low speeds (hence small turn radii) a reduction in longitudinal acceleration capability is not always accompanied by an increase in lateral capability.

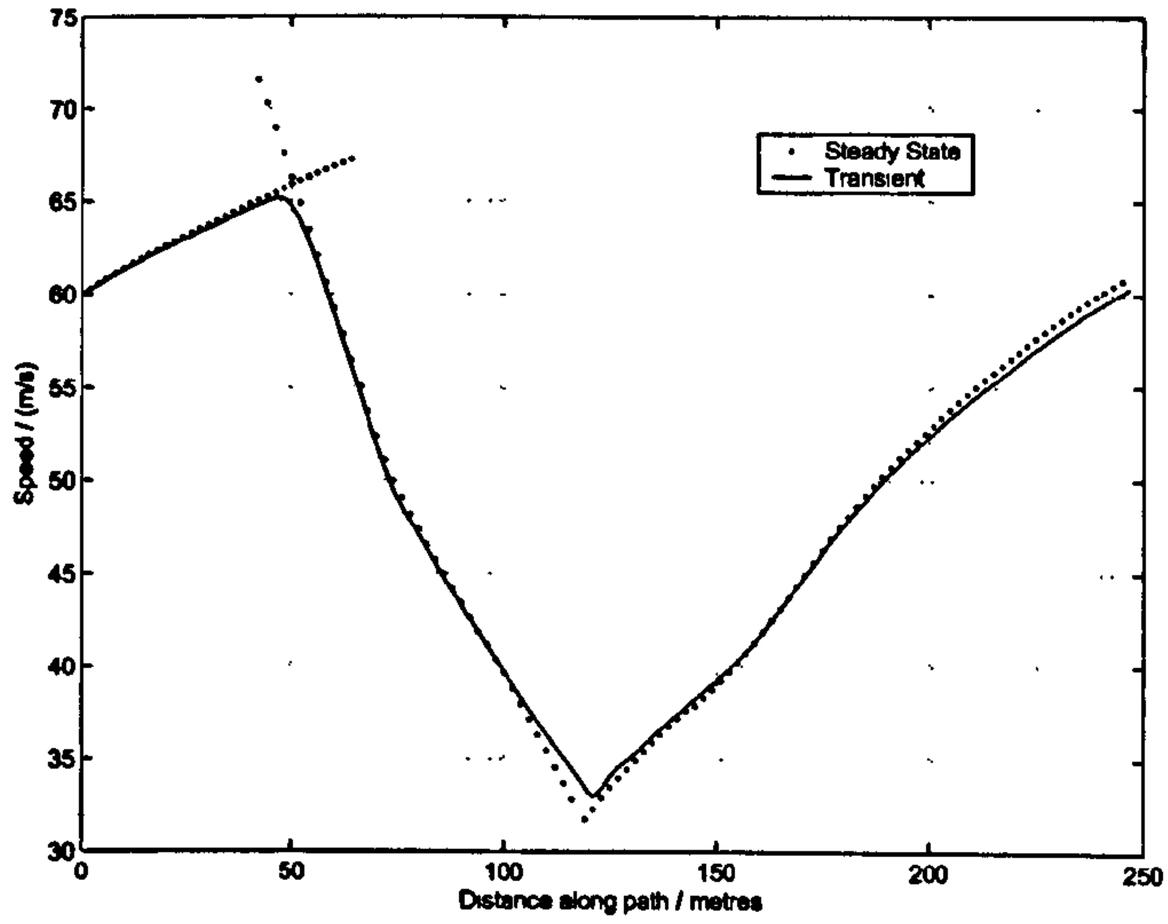


Figure 5.1: Speed profiles for Steady State and Transient manoeuvre time simulators

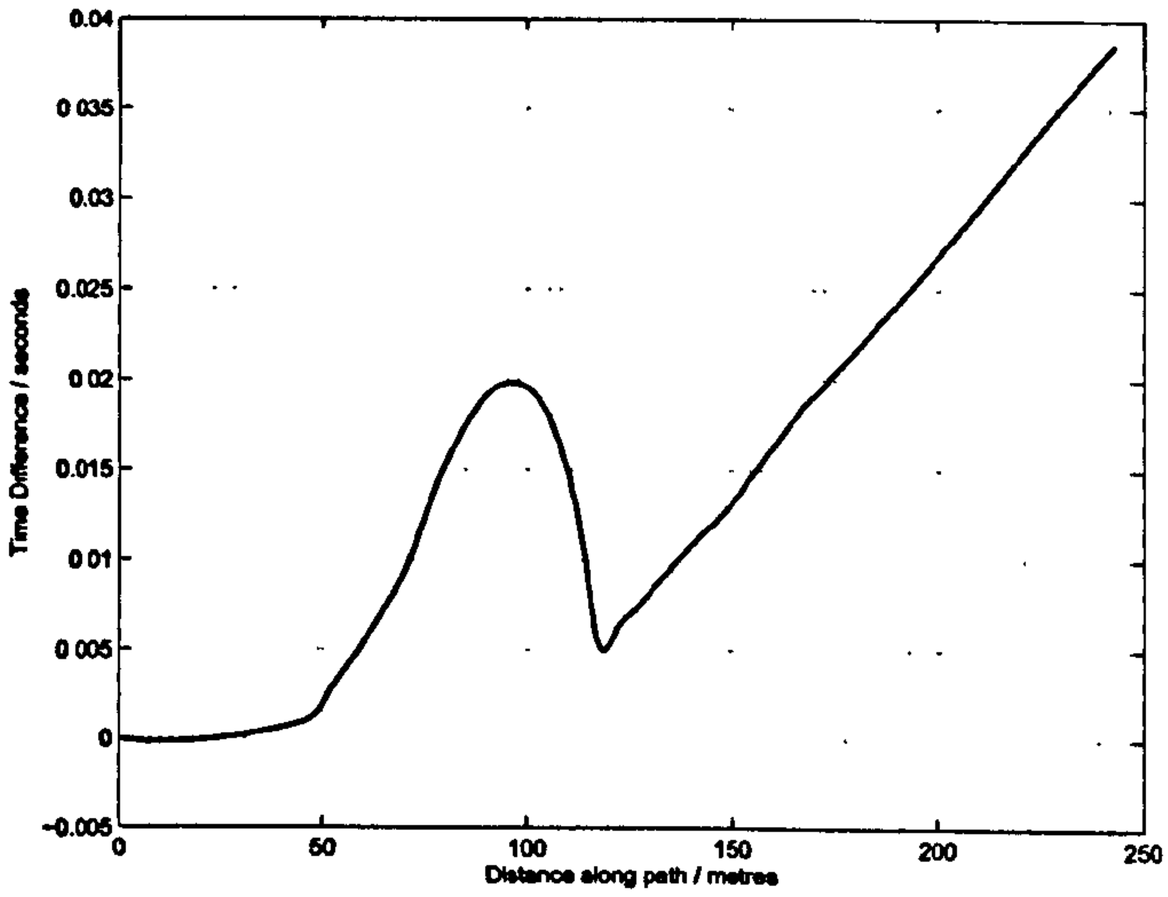


Figure 5.2: Time Difference for Steady State and Transient manoeuvre time simulators

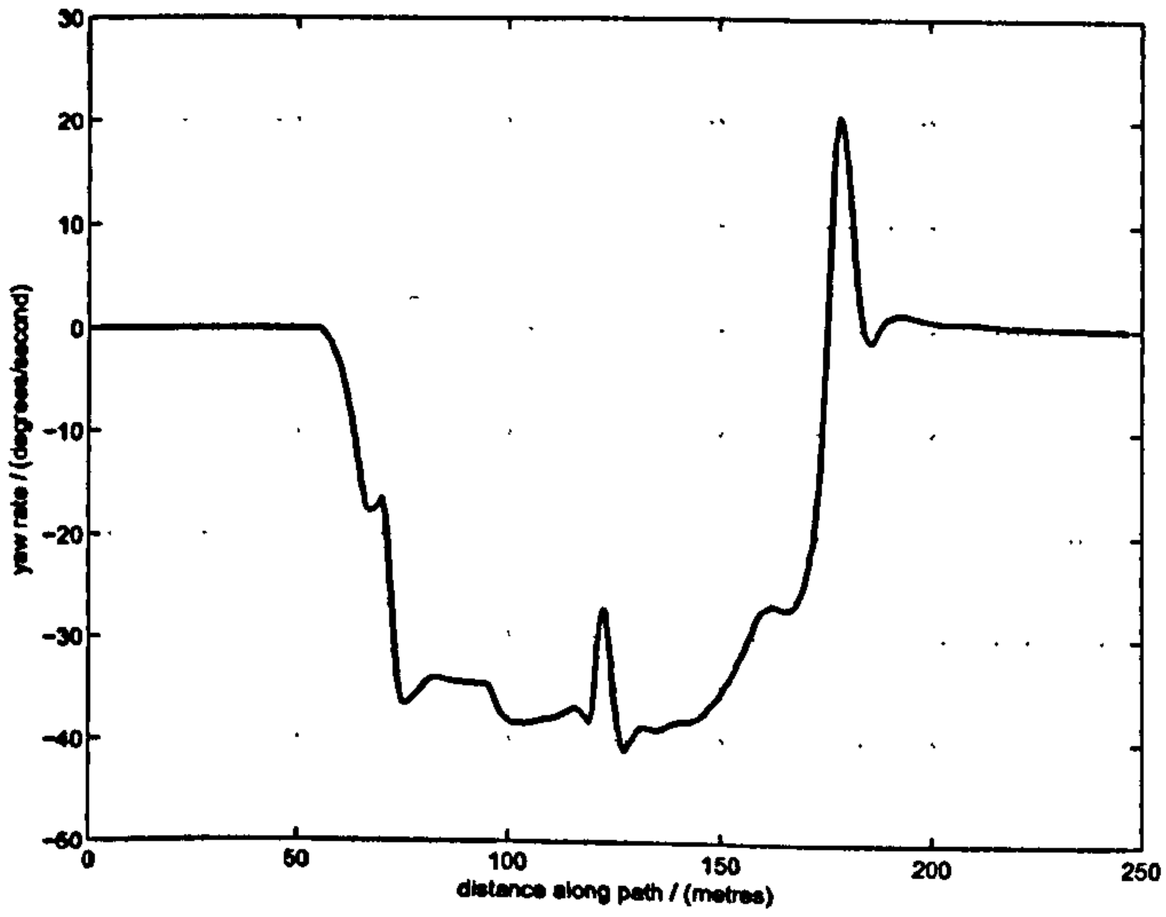


Figure 5.3: Vehicle Yaw Rate from MTM solution

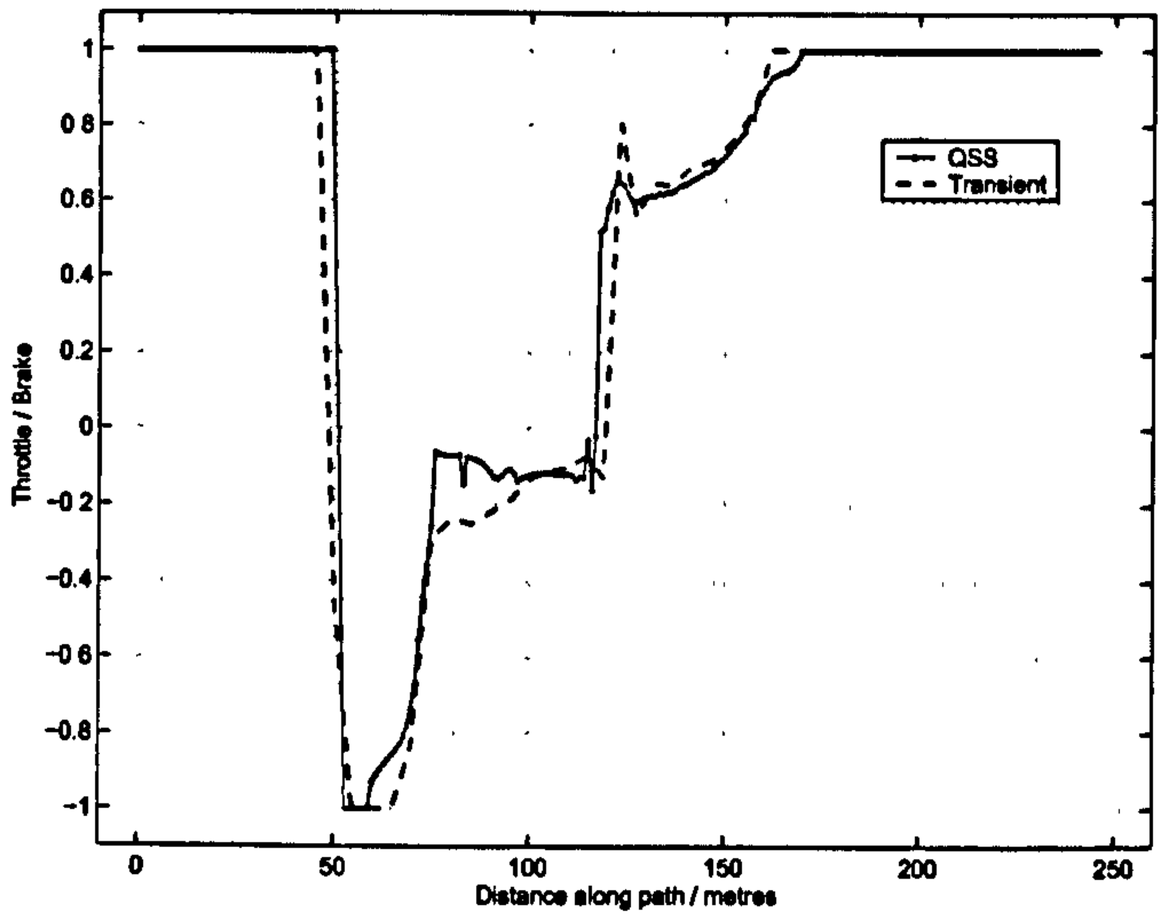


Figure 5.4: Throttle/Brake for Steady State and Transient manoeuvre time simulators

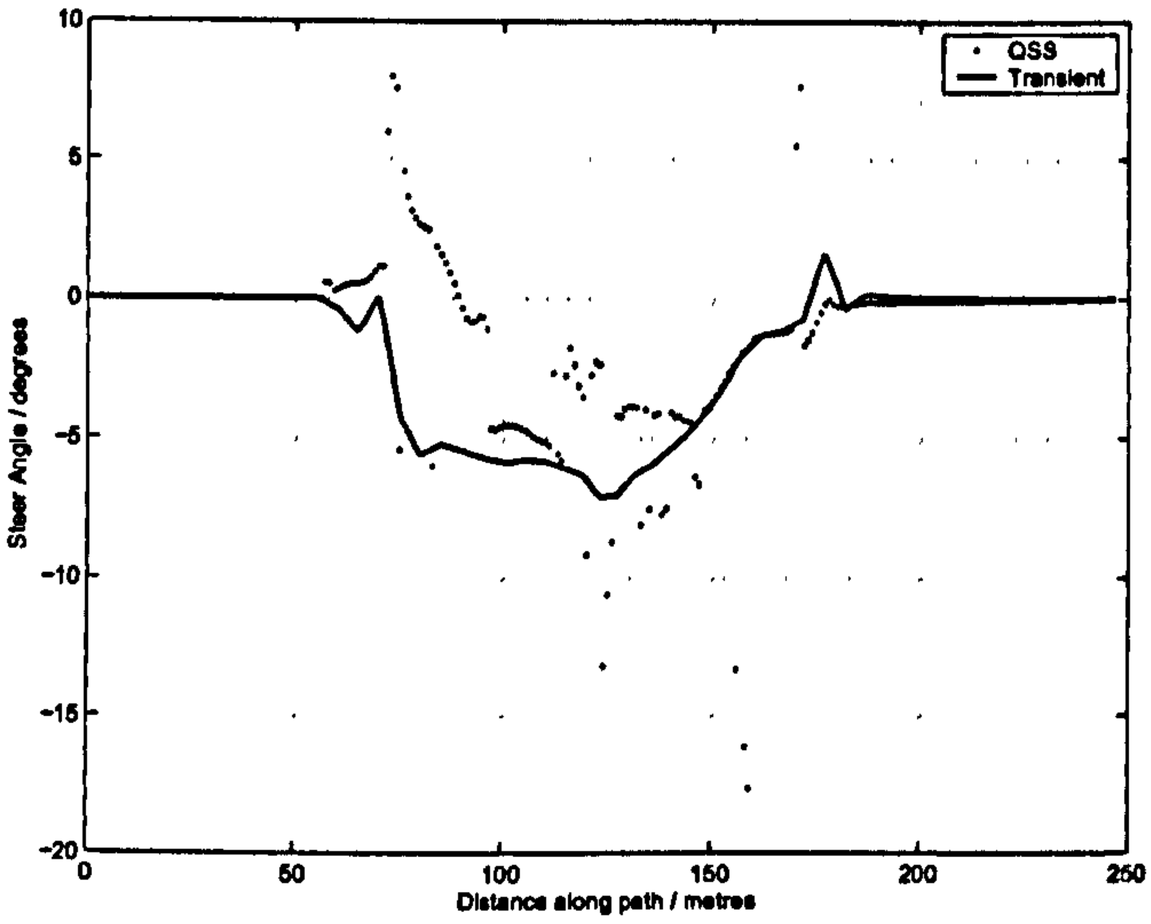


Figure 5.5: Steer Angle for Steady State and Transient manoeuvre time simulators

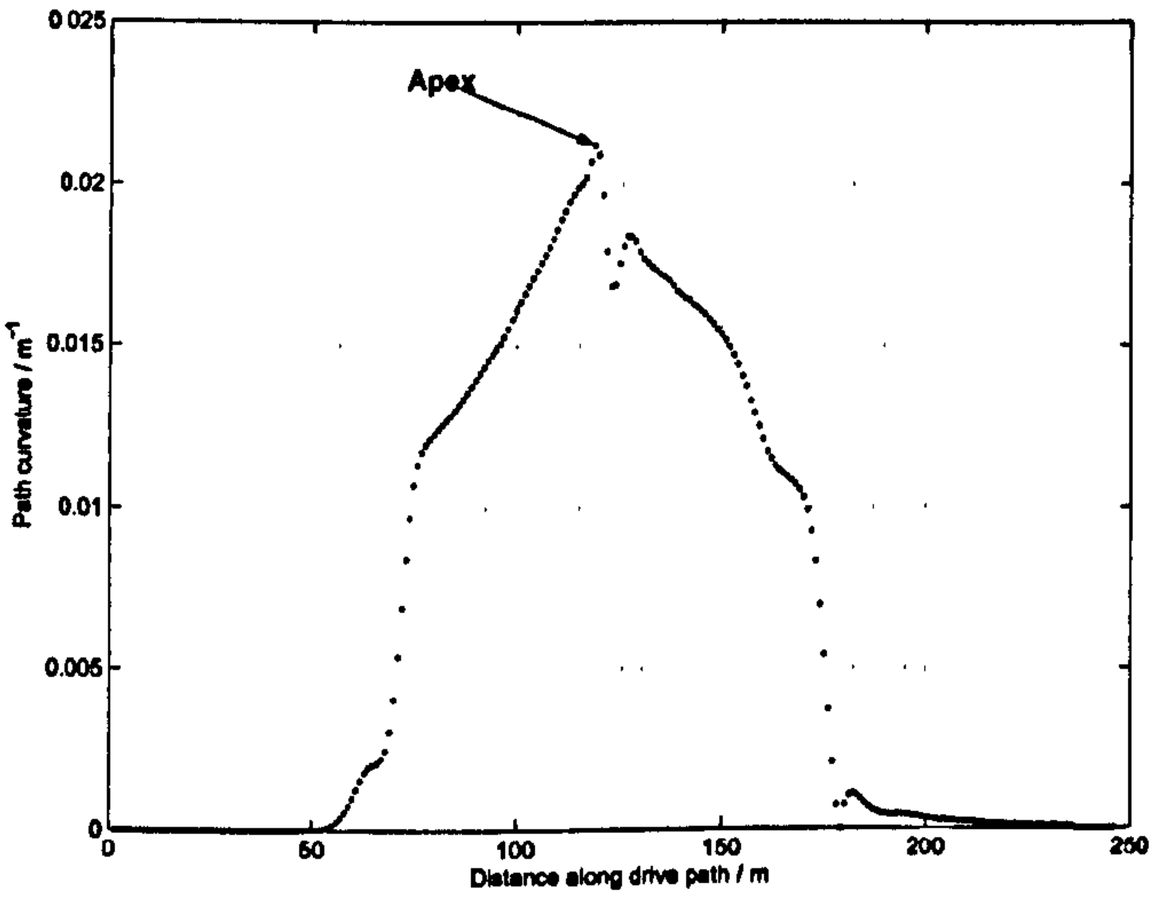


Figure 5.6: Curvature Data for MTM drive line

5.3 Impact on QSS solutions from a change in driven line

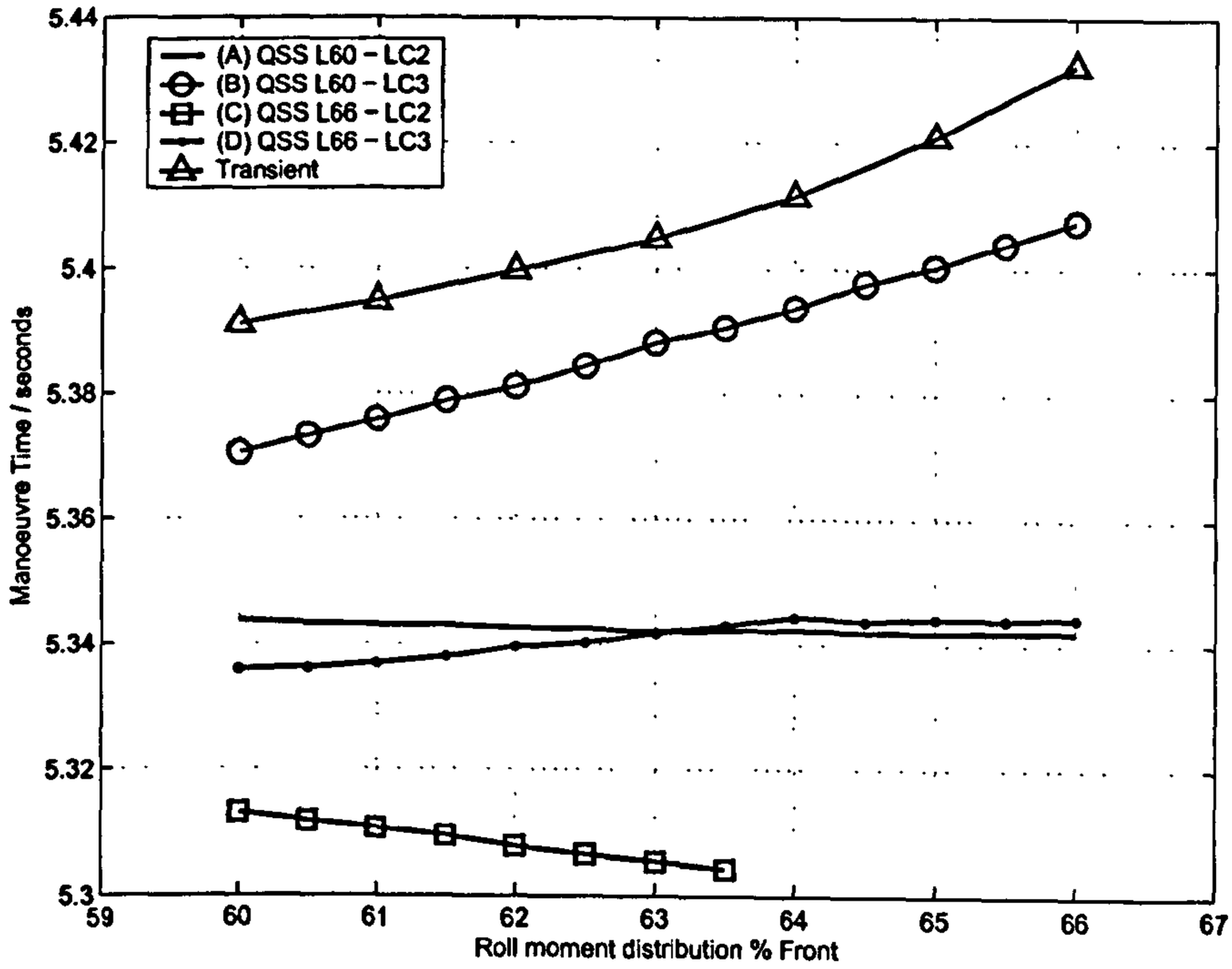


Figure 5.7: Comparison of QSS solutions on differing driven lines

As shown in chapter 4, changes in the value of certain vehicle parameters can result in a significant change to the optimal driven path. As stated in appendix D, the driven path is presented as an input to the QSS manouevre time simulation procedure. In this section we analyse the effect of a change in roll moment distribution on manoeuvre time using the QSS method. We analyse roll moment distribution in the range 60-66% using two driven lines. The driven lines are those produced by the transient method at the end points of this range (60 and 66%). We also show the effect of the 'Apex' longitudinal conditions (as described in Appendix D) on the manoeuvre time. As explained in Appendix D for the QSS procedure, it is necessary to make some assumptions about the vehicle longitudinal acceleration at the Apex. Three alternatives are considered:

- LC1 Longitudinal acceleration of car is unconstrained
- LC2 Longitudinal force generated by loaded rear tyre is zero (maximum lateral capability of tyre exploited)
- LC3 Longitudinal acceleration is set to a pre-determined value (taken from transient simulation)

In all the cases considered method 1 results in a failure of the algorithm to find a solution. This is due to a contraction of the g-g diagram in the region of the maximum

lateral performance. Hence only cases 2 and 3 are considered. Both of the driven lines produce by the MTM exhibit a double 'Apex', that is, the curvature is not monotonic but exhbits two peaks. In the case of the 66% RMD driven line we can ignore the second Apex as the tightening of the corner following the first Apex is not large enough to prevent feasible solutions from being generated. This is true for the 60% RMD solution when using method 3. When using method 2 with 60% RMD line we must treat each Apex separately and perform forward and backward sweeps from each Apex. Hence We consider the four cases as shown below:

- Trace A - 60% RMD driven line using condition 2 (single Apex)
- Trace B - 60% RMD driven line using condition 3 (single Apex)
- Trace C - 66% RMD driven line using condition 2 (double Apex)
- Trace D - 66% RMD driven line using condition 3 (single Apex)

The transient solution is also shown for comparison (Trace E). The results are shown in Figure 5.3. The QSS results show a similar trend to the transient results when the 66% RMD driven line is used and where the longitudinal acceleration at the Apex is taken from the transient solution. It is apparent that when using condition LC2, the trend indicated by the QSS solution is the opposite to that indicated by both the transient solution and the QSS solutions using condition LC3. The speed traces for all three methods on the 60% and 66% driven lines are given in figures 5.3 and 5.8.

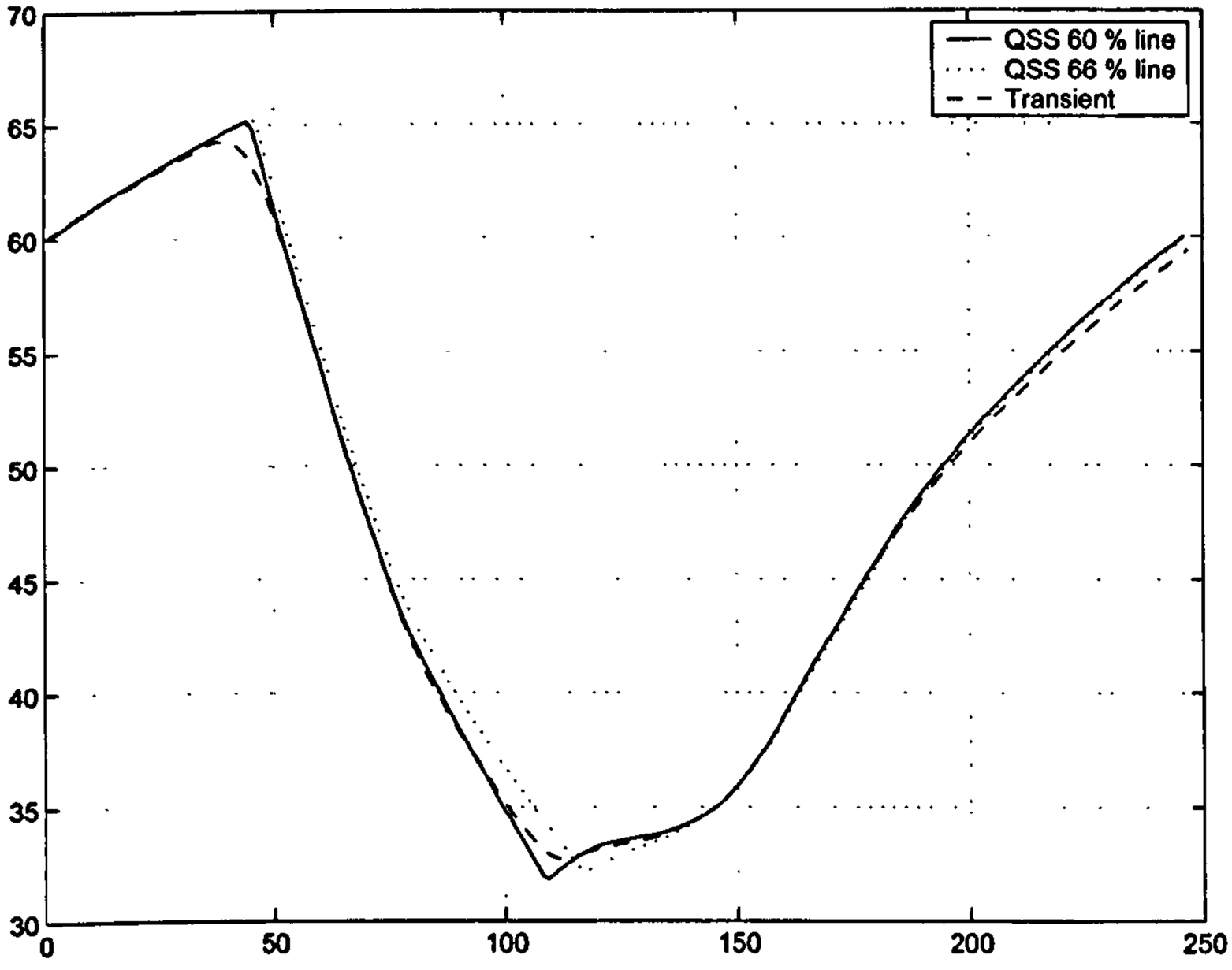


Figure 5.8: Comparison of Speed traces for the 60% forward RMD case

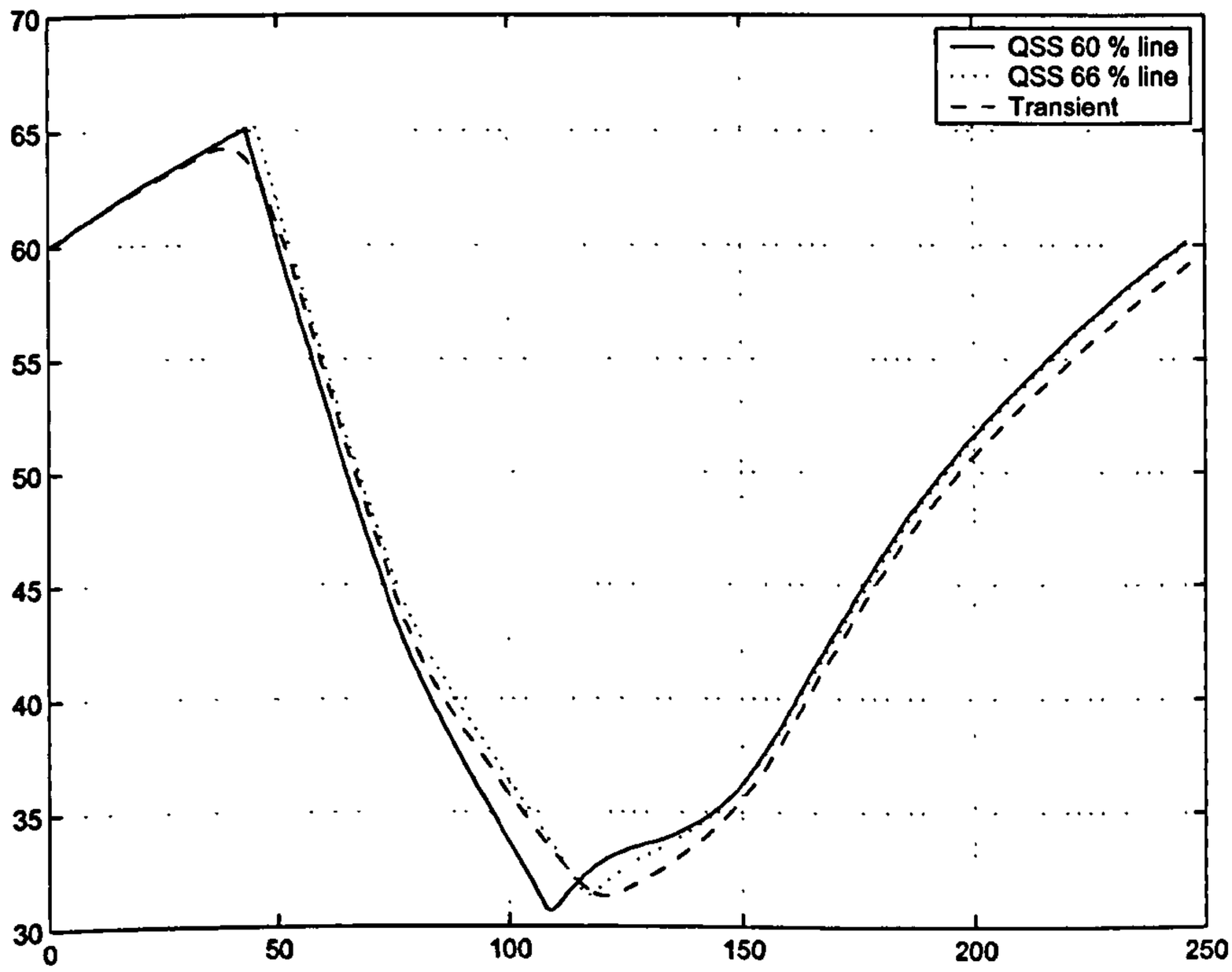


Figure 5.9: Comparison of Speed traces for the 66% forward RMD case

5.4 Review

The speed traces generated by transient and steady state solutions are similar. The differences between them seem reasonable and can readily be explained. The change in racing line associated with a change in vehicle set-up appears to be a significant factor in determining manoeuvre time. This will have an impact on the usefulness of results obtained by the QSS method. The results obtained with the QSS method are very sensitive to the longitudinal acceleration conditions imposed on the solution at the Apex.

Chapter 6

Extending the method for longer manoeuvres

The Farther ahead you look, the smoother and faster you will drive. Practice looking well down the track

Bob Bondurant

6.1 Summary

In this chapter we develop the MTM method described in chapter 2 to allow it so be used with manoeuvres of an arbitrary length. This is achieved using a 'Finite Horizon' approach [3][53]. Use of this approach also allows us to gain some insight into the nature of the driven line and speed profile planning problems. The trade-off between corner entry and corner exit is considered along with the compromises required when corners are coupled in close succesion on the circuit. The results in this chapter are obtained using the QSS vehicle model with set-up QSS/A and tyre set ST/A (see Appendices A and B).

6.2 Definition of 'Finite Horizon' or 'Preview' Method

The distance based method of manoeuvre time optimization is explained in chapter 2. Its adaptation for longer manoeuvres is represented in figure 6.1. The longer manoeuvre is broken down into a series of steps. Each step applies the MTM algorithm to a subset of the manoeuvre. In the example shown in the figure, the first step of the finite horizon procedure applies the MTM algorithm for a manoeuvre between waypoints 1 and 7 from a given initial vehicle state at waypoint 1. On completion of this first step optimization of the control variables at wayline 1 and wayline 2 is complete. In the next step we proceed with a new MTM problem starting at wayline 2 and ending at wayline 8. The controls at wayline 2 are

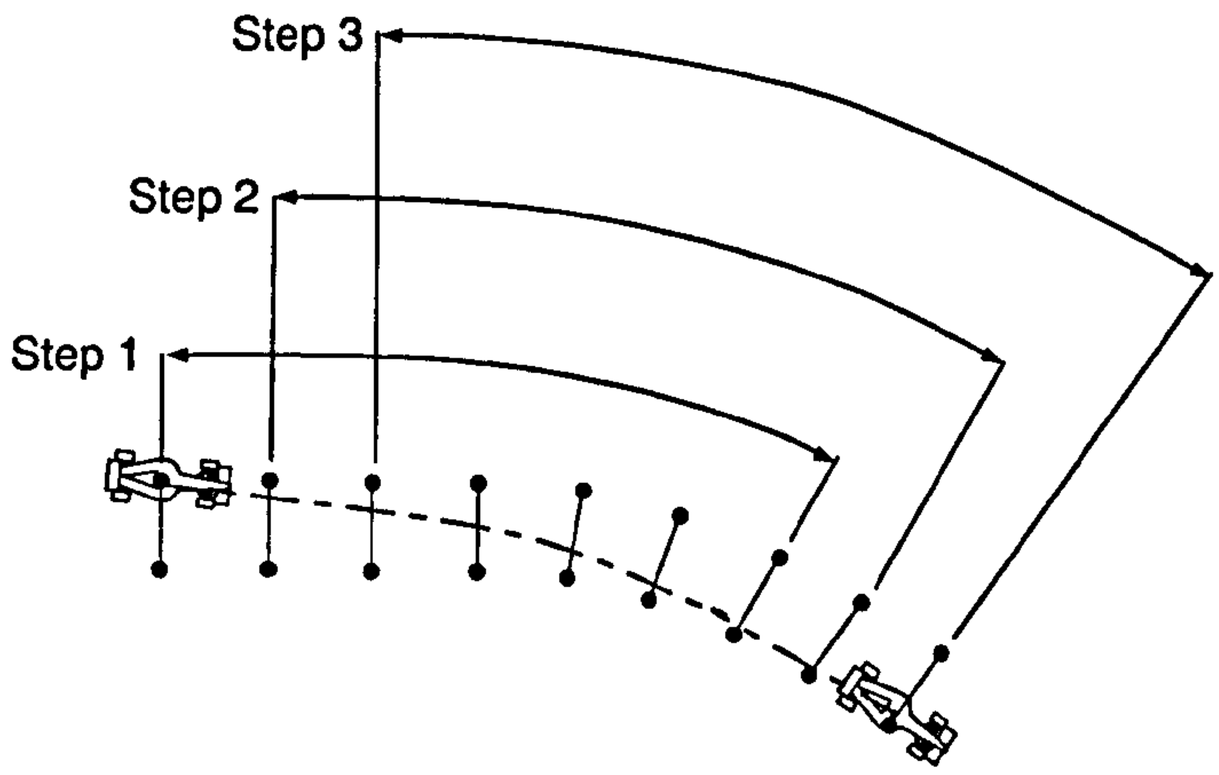


Figure 6.1: Finite Horizon Procedure

fixed at the optimized values found in step 1. The initial vehicle state for step 2 is taken from the vehicle state on crossing waypoint 2 in the optimal trajectory found in step 1. On completion of step 2 optimization of the control values at waypoint 3 is complete. Step 3 and subsequent steps proceed in a similar fashion to step 2. The procedure terminates when all the control values in the manoeuvre have been optimized.

At each step of the finite horizon procedure the MTM algorithm is used to compute optimal control values for a vehicle state trajectory between two given waylines. At the end of the step, the control values at the first wayline are retained and the control values at subsequent waylines will be re-optimized in the next step. Hence, the retained control values at the first wayline are optimized taking into account a 'preview distance' or 'finite horizon' to the last wayline. This procedure is described more formally in the following section.

6.2.1 Definition of Finite Horizon Algorithm

In the general case we wish to compute optimal control values hence an optimal vehicle trajectory for a manoeuvre consisting of N waypoints. We assume that the manoeuvre is too large to be optimized by the MTM algorithm alone for the reasons discuseed in chapter 2. The finite horizon algorithm breaks the larger manoeuvre down into a series of steps. Each step applies the MTM algorithm to a subset of the larger manoeuvre. Step j of the finite horizon procedure computes a time optimal trajectory between wayline n_j^{start} and n_j^{end}.

6.2.1.1 Initial state

In the MTM algorithm, computation of the objective function consists of solving an initial value problem in the differential equation describing the vehicle model from a given 'start' wayline to a given 'end' wayline. The initial value (the vehicle state at the beginning of the manoeuvre) is given. The output of the computational procedure is the state trajectory traversed by the vehicle sampled discretely over time. Each sampled state trajectory includes the vehicle states at which the vehicle centre of gravity crosses each wayline in the manoeuvre. Let the vehicle state on crossing waypoint n be given by $\bar{m}(t_{Ln_j})$ where t_{Ln_j} is the crossing time. On termination of the MTM algorithm we have an optimal state trajectory. Let $\bar{m}_{j*}(t_{Ln_j})$ be the state on crossing waypoint n in the optimal trajectory computed during step j of the finite horizon algorithm. At each step of the finite horizon algorithm the initial vehicle state is set equal to the state on crossing wayline n_j^{start} in the optimal trajectory computed in step $j-1$. This is given by $\bar{m}_{(j-1)*}(t_{Ln_{j-1}^{start}})$. In the case of step 1 the initial state is assumed given.

6.2.1.2 Vehicle Control Inputs

Let $\bar{u}_{j*}^n$ be the vector of control values at waypoint n associated with the optimal trajectory computed during step j of the finite horizon procedure. At each step of the finite horizon algorithm the control values at waypoint n_j^{start} are fixed at the optimal values found in the previous step $\bar{u}_{(j-1)*}^{n_j^{start}}$. At each step of the algorithm one new vector of control points is introduced (at the new end wayline). Initial values for this and all other controls for the MTM at step j are typically set to random values. Alternatively, for waypoints that have meaningful control values from a previous step, we have the option to retain the values computed at the previous step as the initial values for the next step. The decision to 're-cycle' previous control values or to replace them with new randomly generated values is a compromise. We would expect re-cycled values to be closer to optimal and therefore would expect the MTM algorithm to converge more quickly. On the other hand, the use of new random values should allow a wider search of the solution space.

In the case of step 1 the control vector at wayline n_1^{start} may be given or it may be optimized by including it in the optimizer independent variables for this step.

6.2.1.3 MTM algorithm

The MTM algorithm is applied at each step as follows. Let $\bar{c}_j^n$ be the vector valued function defining the constraints at waypoint n. For $n > n_j^{start}$ and $n \leq n_j^{end}$ find $\bar{u}_j^n$ that minimise the time taken to travel between n_j^{start} and n_j^{end} subject to satisfaction of the constraint functions $\bar{c}_j^n$. The initial vehicle state (at waypoint n_j^{start}) and the control vector $\bar{U}_j^{n^{start}}$ are taken from the optimized trajectory generated in the previous step:

$$\bar{m}_{j*}(t_{Ln_j^{start}}) = \bar{m}_{(j-1)*}(t_{Ln_j^{start}}) \tag{6.1}$$

$$\bar{u}_j^{n_j^{start}} = \bar{u}_{(j-1)*}^{n_j^{start}} \tag{6.2}$$

6.2.1.4 Termination

At each stage of the algorithm we advance as follows:

$$n_j^{start} = n_{j-1}^{start} + 1 \tag{6.3}$$

$$n_j^{end} = n_{j-1}^{end} + 1 \tag{6.4}$$

Termination of the finite horizon procedure is achieved when the vehicle control values associated with all N waypoints have been optimized. When a manoeuvre consists of a finite length segment of a racetrack we will 'run out' of preview when $n_j^{end} > N$. Under these circumstances two options are available:

- Reduce the preview distance for the affected points such that $n_j^{end} \leq N$
- add sufficient waylines beyond wayline N to maintain the preview (for example, add a straight section at the end of the manoeuvre)

For a racing circuit, the manoeuvre is effectively of infinite length so this problem does not arise.

6.2.1.5 Notes

The distance between waylines n_j^{start} and n_j^{end} defines the distance to horizon or preview used to optimize the control vector at wayline n_j^{start}. This is given by $D^n_{preview}$. Each control point is optimized taking into account a preview distance $D_{preview}$. The effect of preview distance on manoeuvre time is studied in the remainder of this chapter. The finite horizon algorithm is easily modified to allow the preview distance to be defined for each wayline in the event that there are differing preview requirements at various stages of a manoeuvre.

In the description given above we advance by a single wayline at each step. The choice of wayline spacing interval also effectively specifies the bandwidth of the vehicle control system. We may wish to have a close wayline spacing to achieve a high control bandwidth but for reasons of computational efficiency to advance by more than one wayline at each step of the finite horizon procedure.

The time based MTM method (chapter 2) can be extended for arbitrarily long manoeuvres in a similar fashion.

6.3 Influence of preview time/distance on driven line and manoeuvre time

In this section we show the effect of preview distance on manoeuvre time and driven line. Figure 6.2 shows an extended version of the single 90° corner. The midpoint of the corner is marked 'Apex' in the figure. This manoeuvre was optimized using the preview algorithm using preview distances in the range 100 to 350m. The resulting manoeuvre times are shown in figure 6.3. The manoeuvre time can be seen to decrease as the preview distance increases. The rate of change of manoeuvre time with respect to preview distance decreases significantly as the preview distance reaches 220m. The wayline at the mid corner or 'Apex'

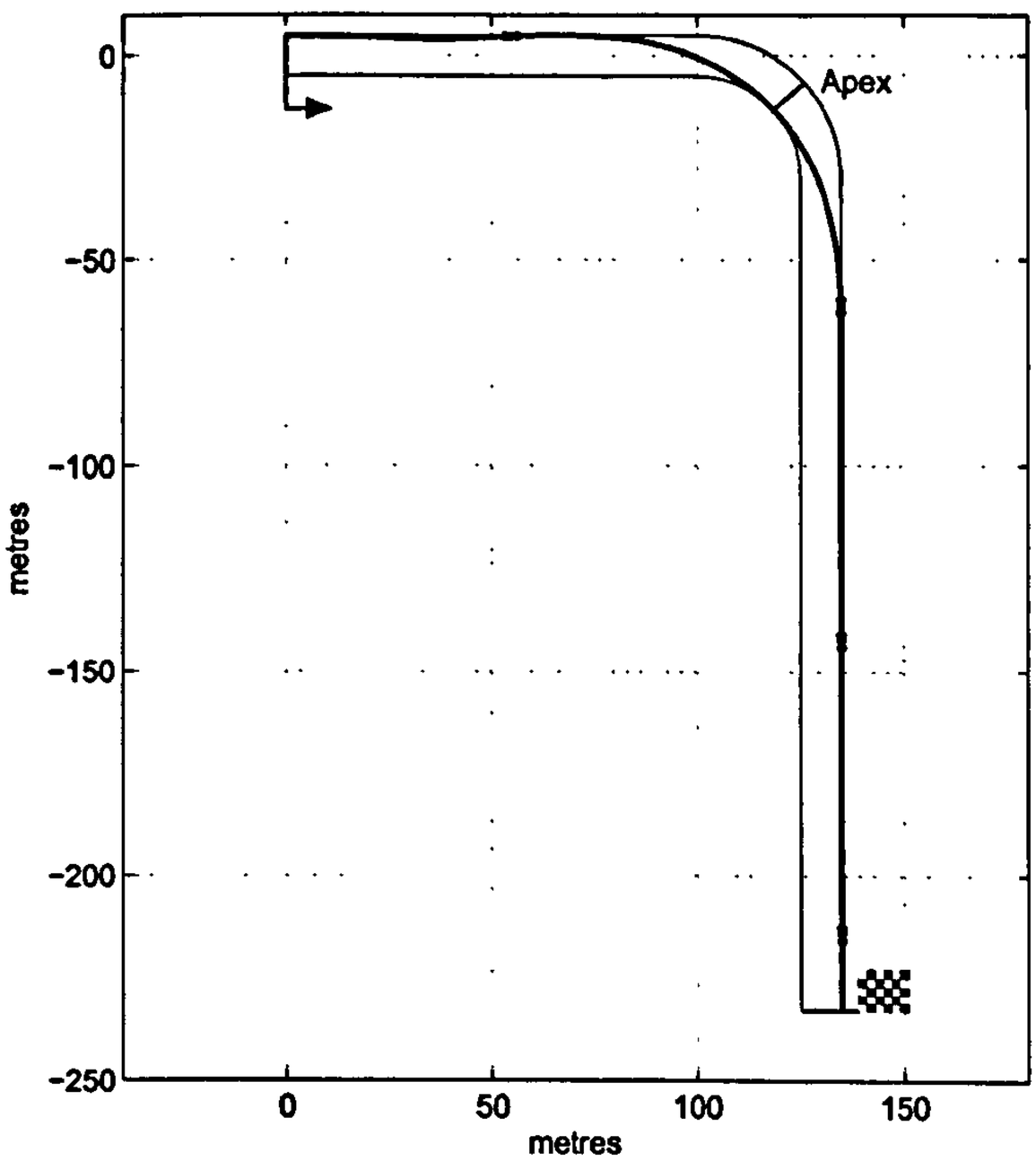


Figure 6.2: Single 90° corner Manoeuvre

point is marked in figure 6.2. Figure 6.4 shows the split manoeuvre times, a) the time required to reach the apex and b) the time required to complete the manoeuvre from the apex. Each split time is shown as the time difference relative to the 100m preview case. The split times show that for shorter preview distances, the car arrives at the Apex in less time but then needs more time to complete the manoeuvre. This 'greedy' approach can be explained by the lack of available preview. This effect is also shown in figure 6.5 where the vehicle speed on crossing the wayline at the apex is higher for shorter preview distances with a corresponding reduction in final speed. Figure 6.7 shows the optimized trajectories produced by some of the steps in the finite horizon algorithm for a preview distance of 120m. This also highlights the short-sighted approach produced by the lower preview values. Finally figure 6.14 shows a 3d view of the car speed and throttle/brake controls overlaid on the manoeuvre. The manouevre effectively begins at the braking point, the points approximately 200 and 250m beyond the braking point are marker in the figure. This helps to give an idea how 'far ahead' the optimization process needs to 'see' when choosing the braking point for this manoeuvre. It is also worth noting that a certain minimum preview distance is required simply to allow the car to remain on a feasible trajectory (i.e. stay on the race track). In the case of this corner with the given vehicle initial state the minimum preview is 100m.

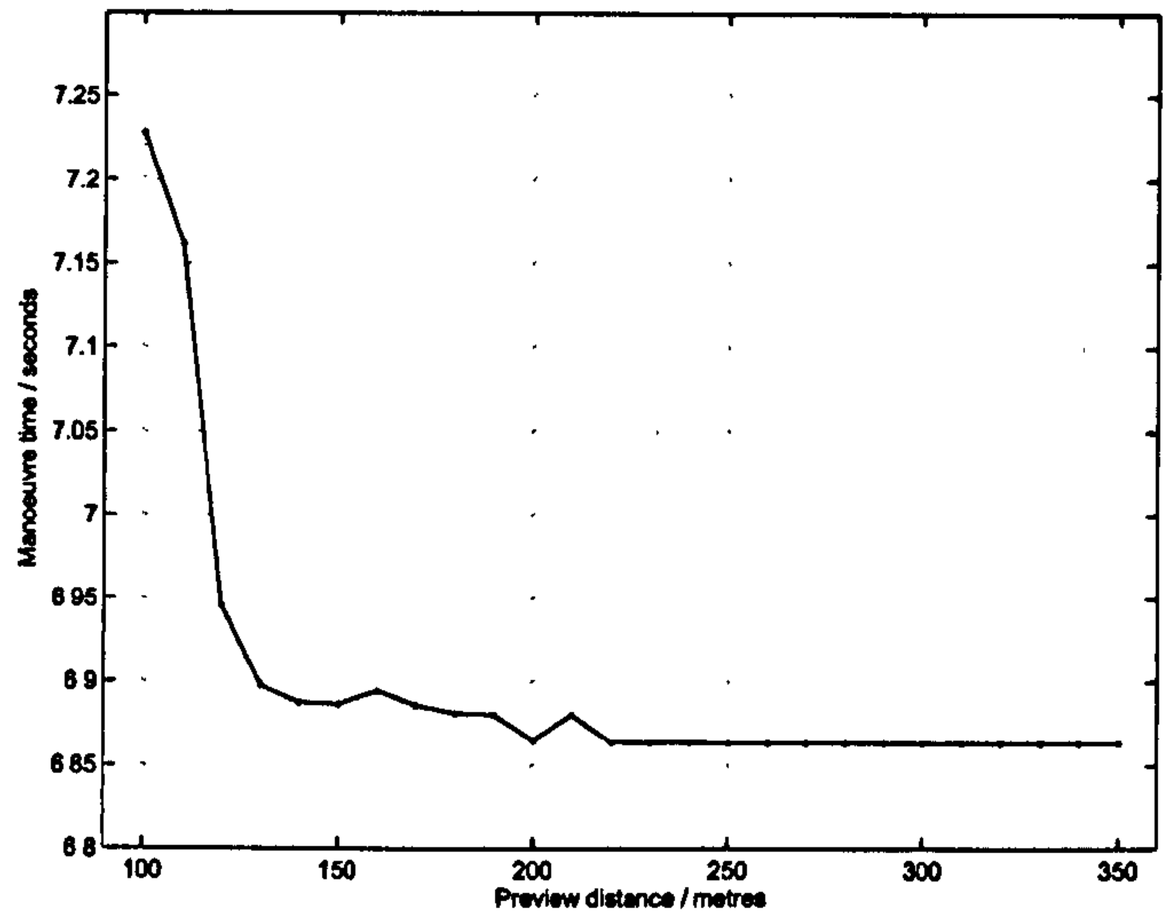


Figure 6.3: Manoeuvre time effect of preview distance

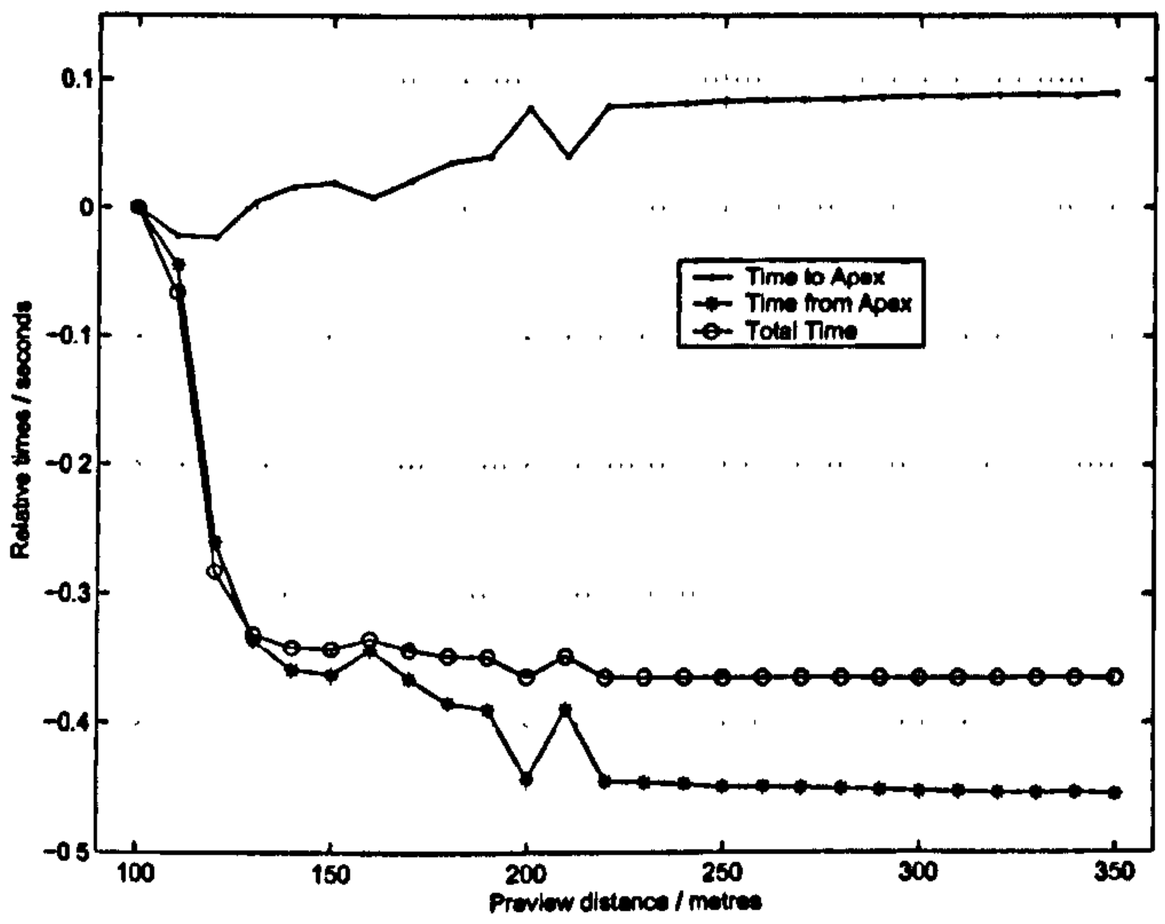


Figure 6.4: Relative Manoeuvre times including split times before and after Apex

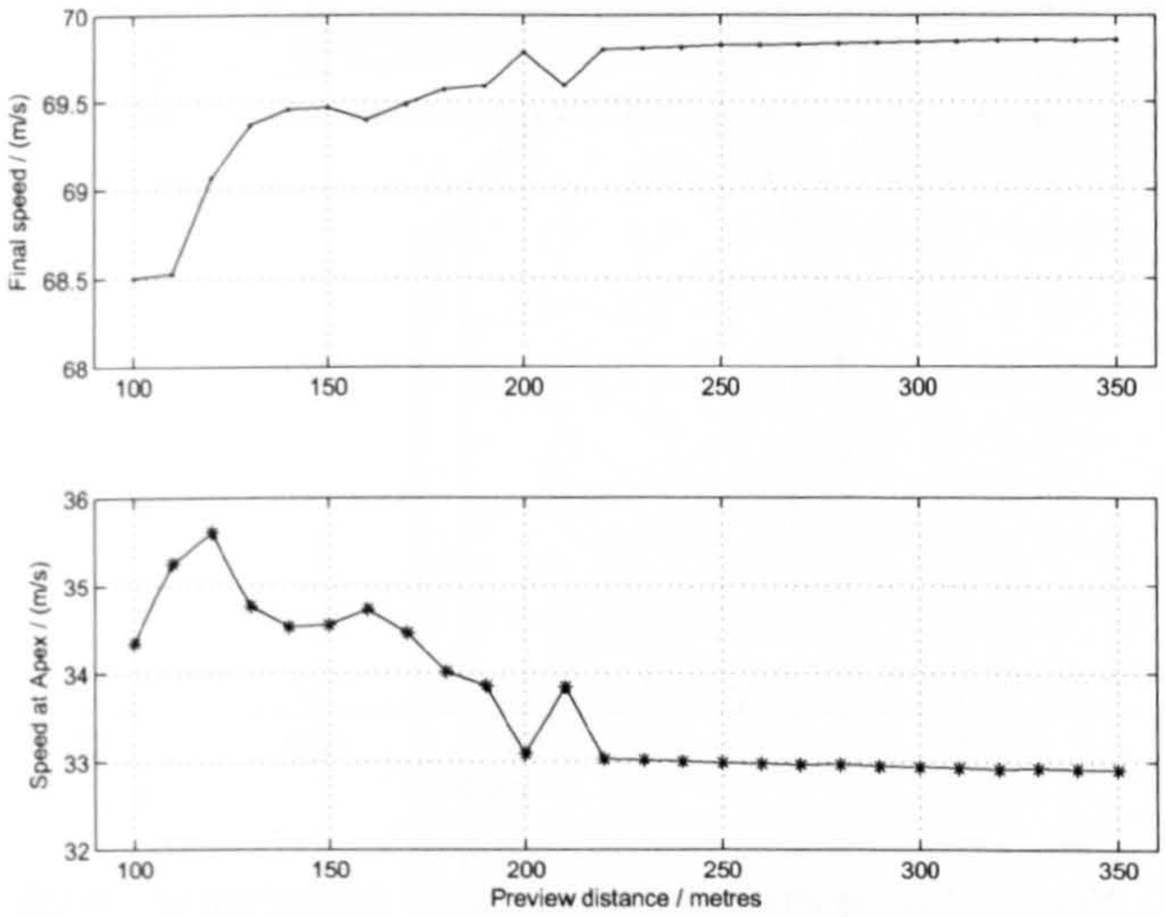


Figure 6.5: Apex and Final Speeds

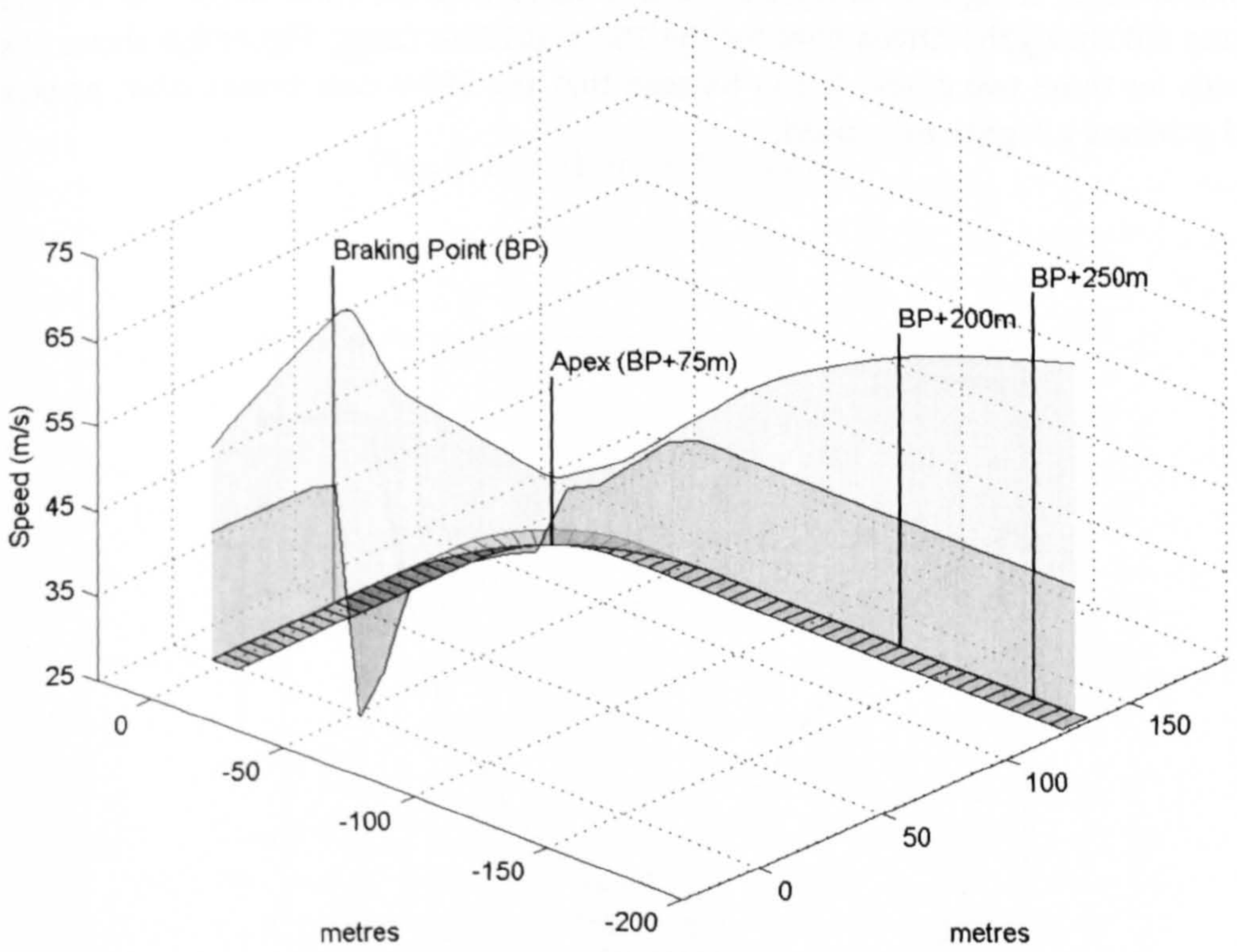


Figure 6.6: Single 90° corner 3D view

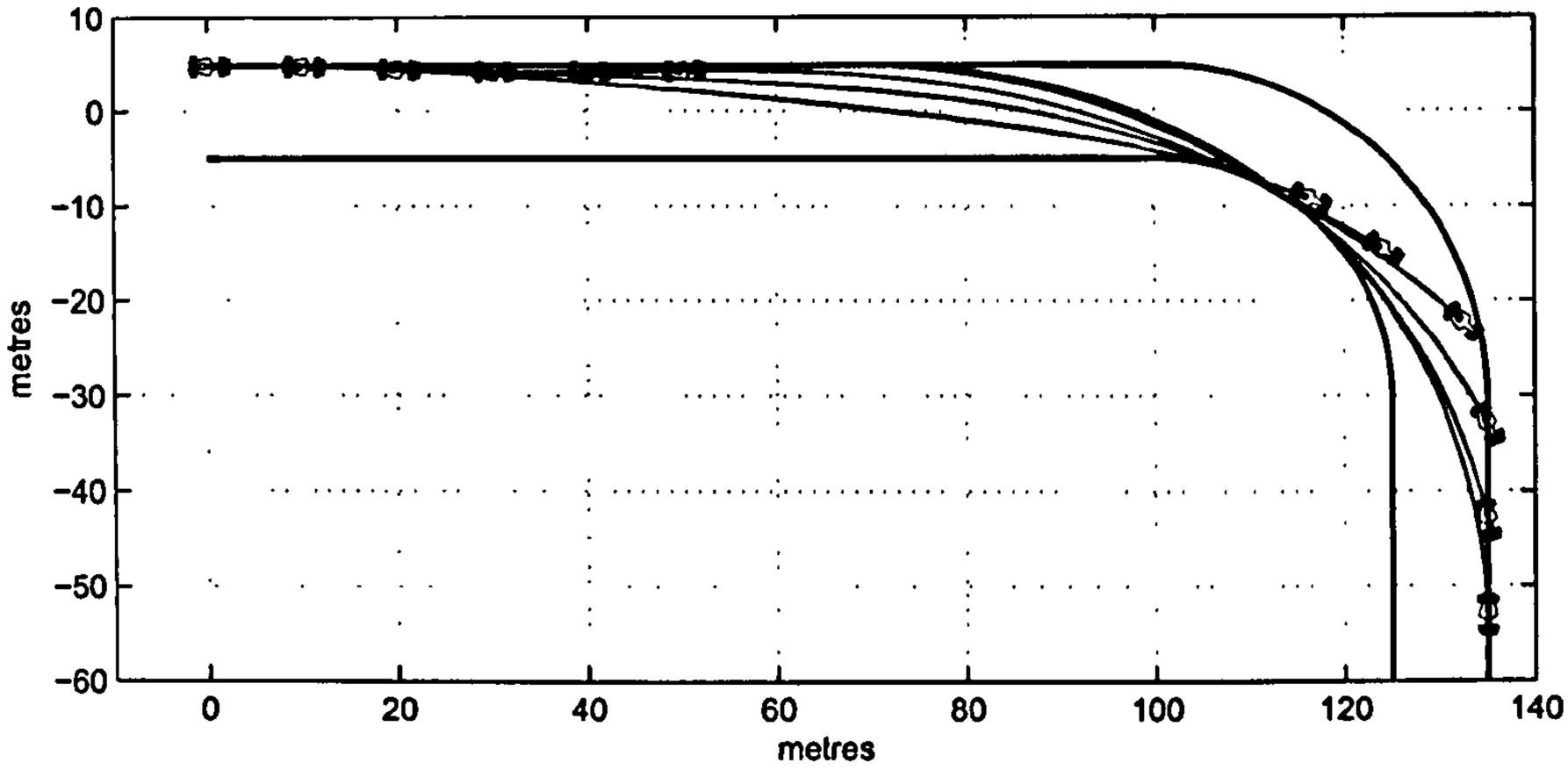


Figure 6.7: Single 90° corner intermediate trajectories with 120m preview

6.3.1 Hairpin

A similar procedure was carried out for the hairpin manoeuvre shown in figure 6.8. Figure 6.10 shows the relationship between manoeuvre time and preview distance. This curve has a similar shape to fig 6.3. In the case of this manoeuvre the curve flattens at around 200m. Figure 6.8 shows the driven lines for the 150 and 290m cases. Figure 6.9 shows the driven speeds for these two cases. It can be seen that the 290m case brakes later, apexes slower and achieves a higher exit speed.

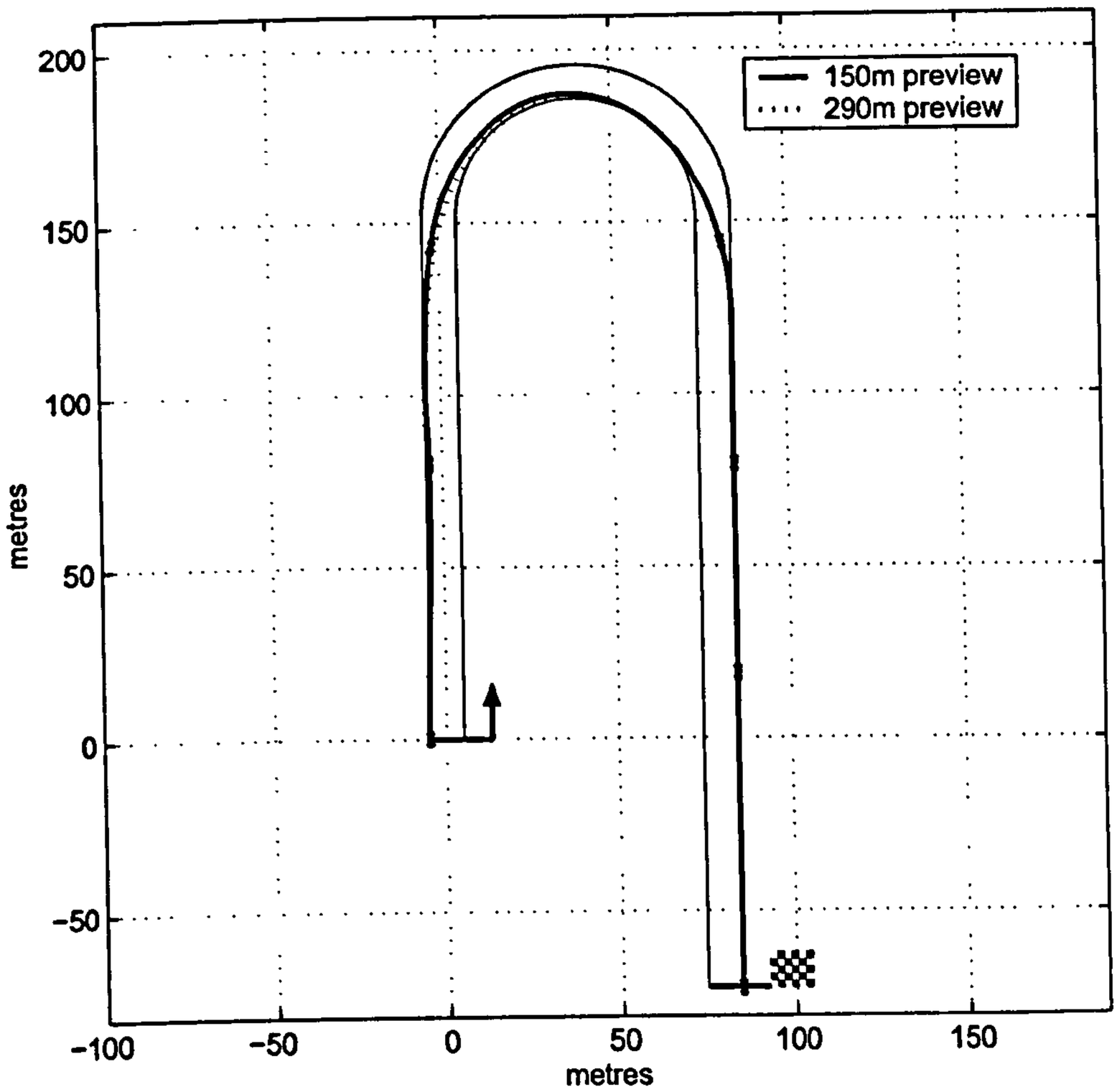


Figure 6.8: Hairpin Manoeuvre

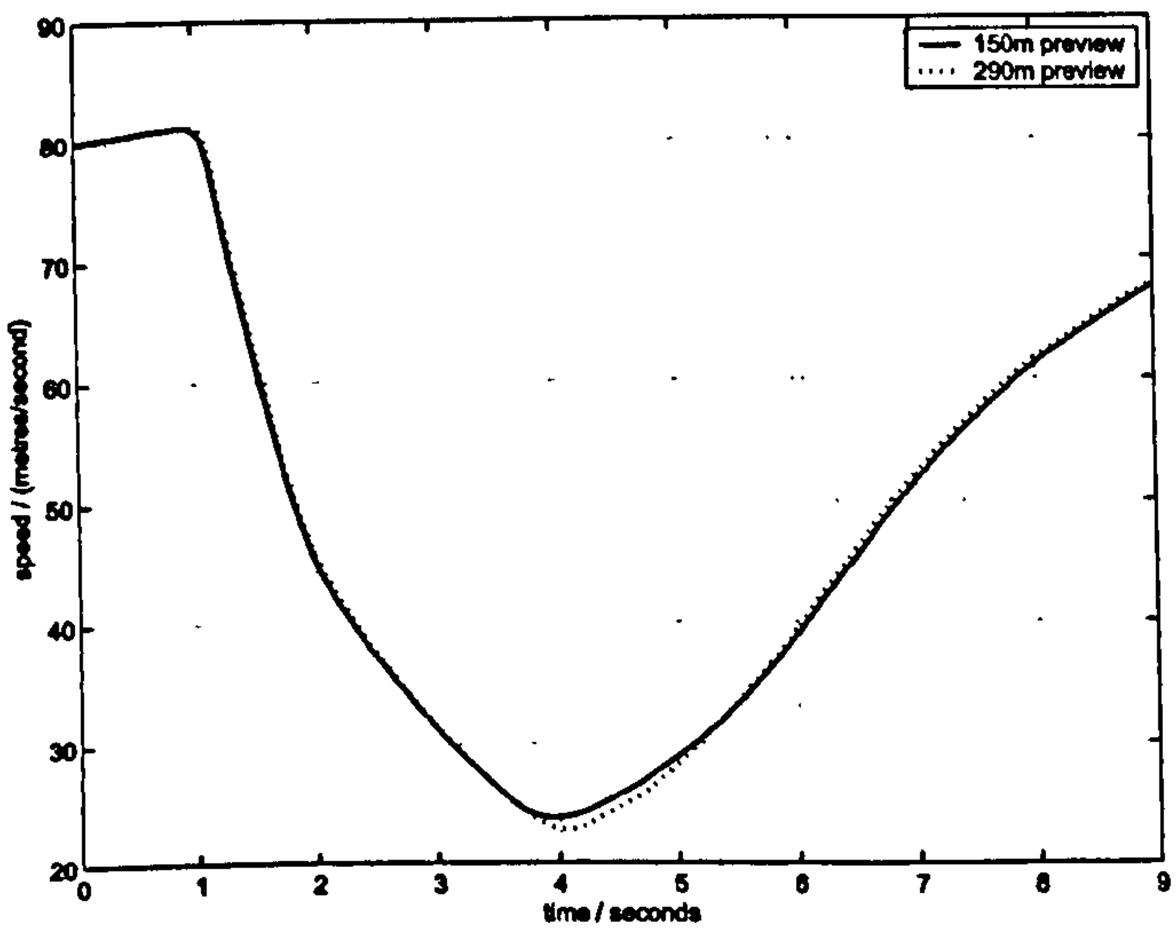


Figure 6.9: Hairpin Speed Comparison

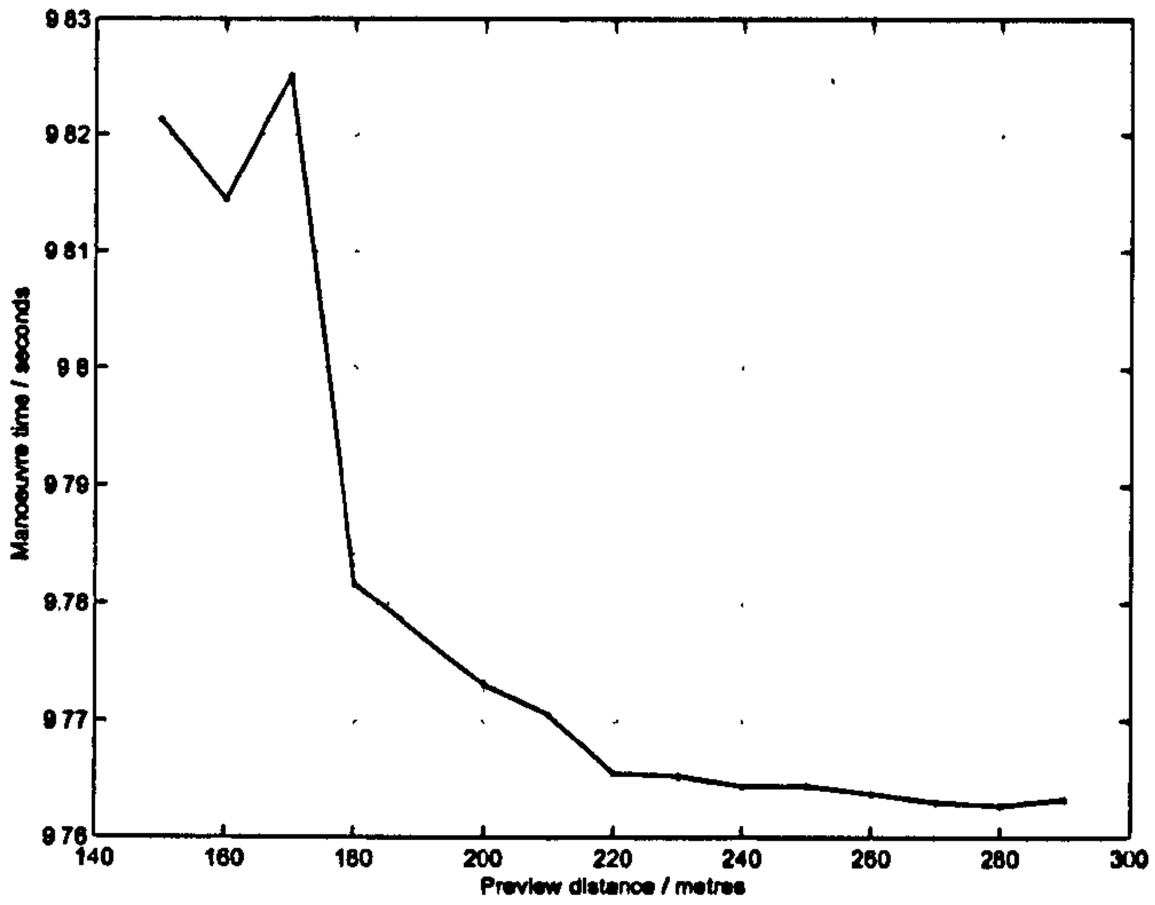


Figure 6.10: Hairpin manoeuvre time effect of preview distance

6.3.2 Coupled Corners

In this section we consider two corners in close succesion as shown in figure 6.11. The influence of preview distance on manoeuvre time is given in figure 6.12 and the corresponding final speed is given in 6.13. The time curve begins to flatten at around 160m and is quite flat by 210m. The speed curve continues to show increases in final speed at 300m preview indicating that further exit speed benefits would be available at higher preview distances. Figure 6.14 shows a 3d representation of the throttle/brake controls and speed profile overlaid on the manoeuvre. This allows us to see that the initial braking point A is approximately 150m before the apex point of the second corner D and 200m before the point at which the car is heading straight on the exit of the manoeuvre.

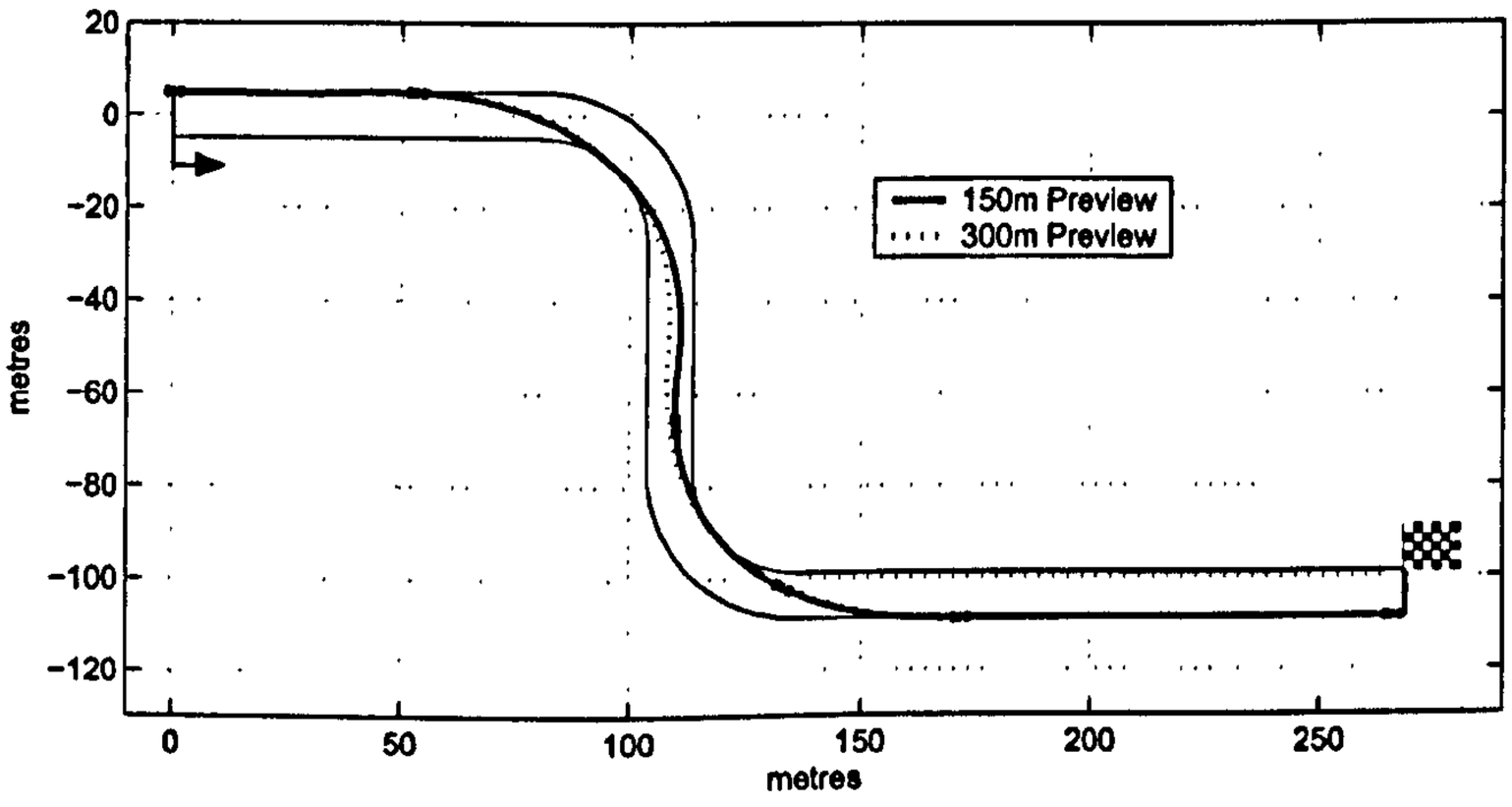


Figure 6.11: Coupled Corner Manoeuvre

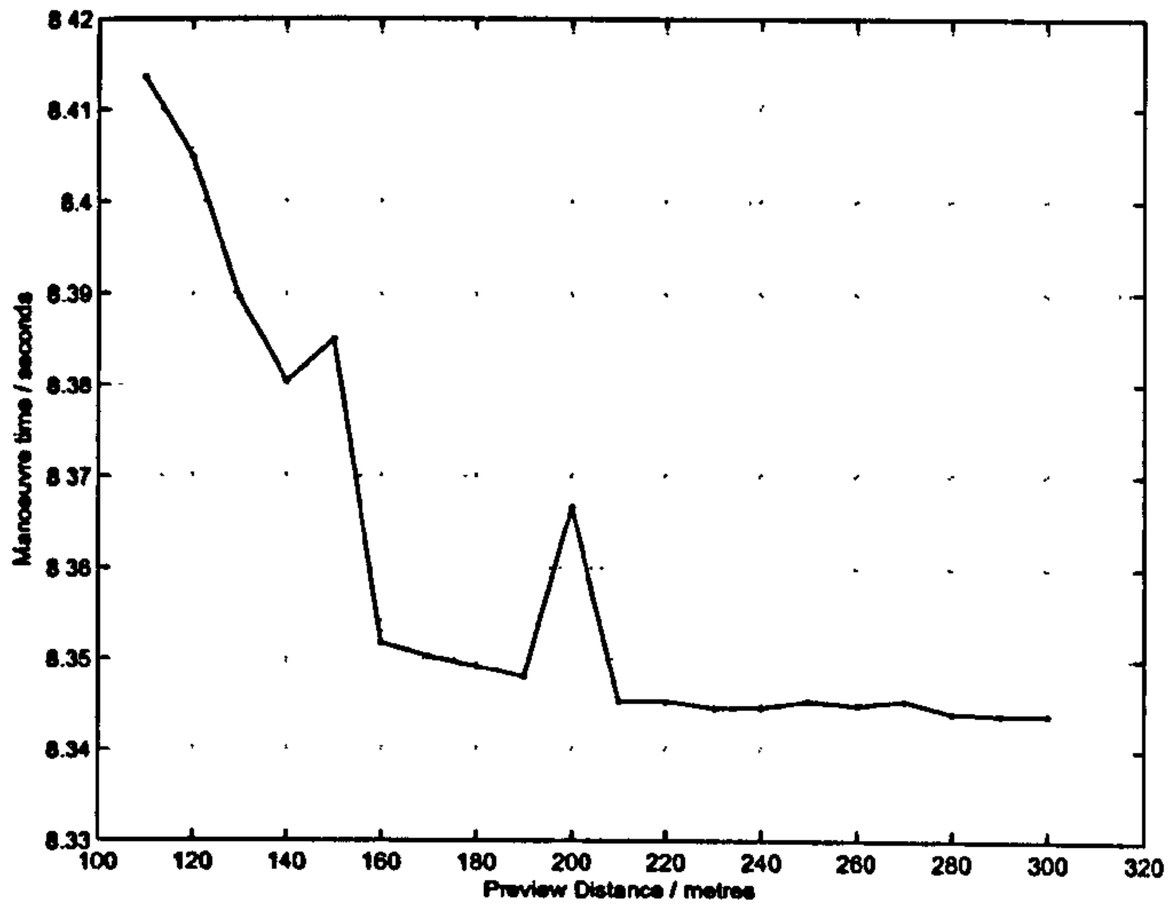


Figure 6.12: Coupled Corner Manoeuvre time effect of preview distance

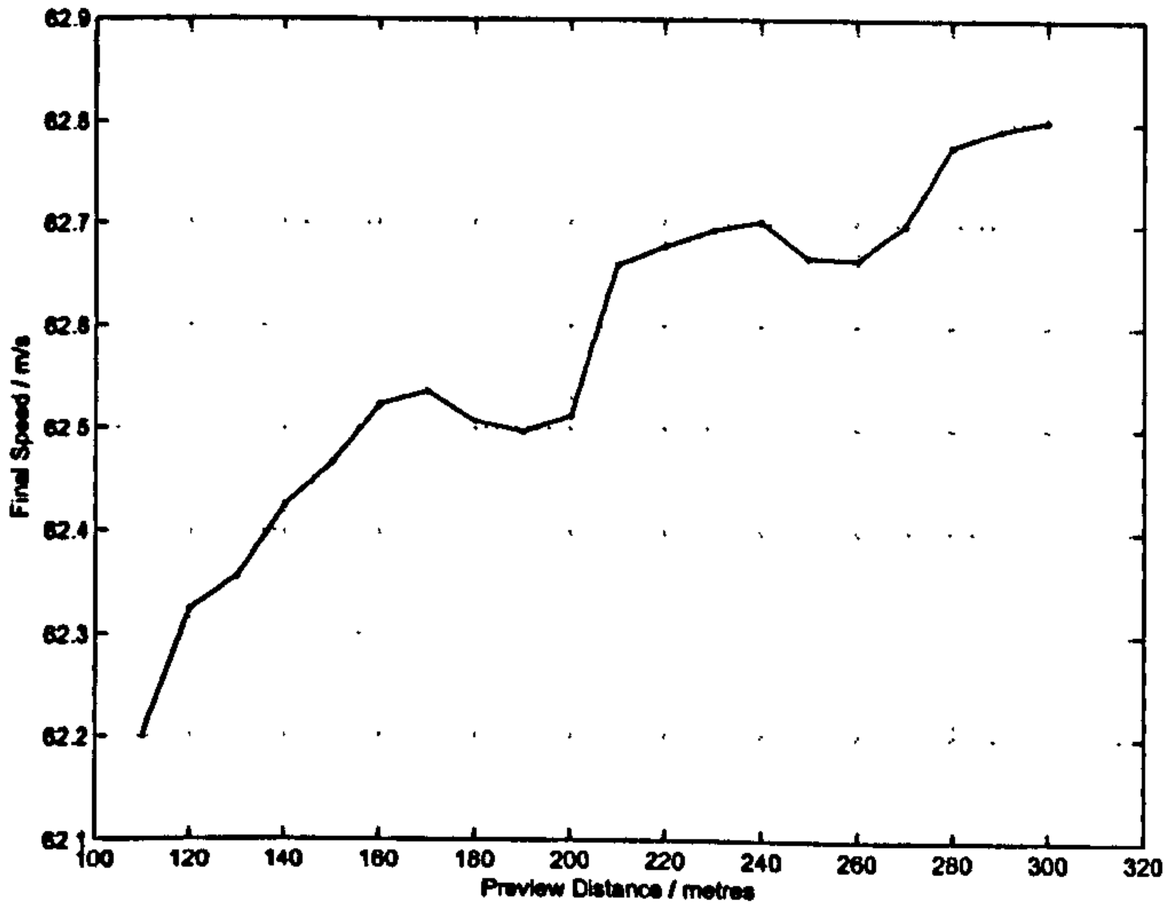


Figure 6.13: Coupled Corner Final speed effect of preview distance

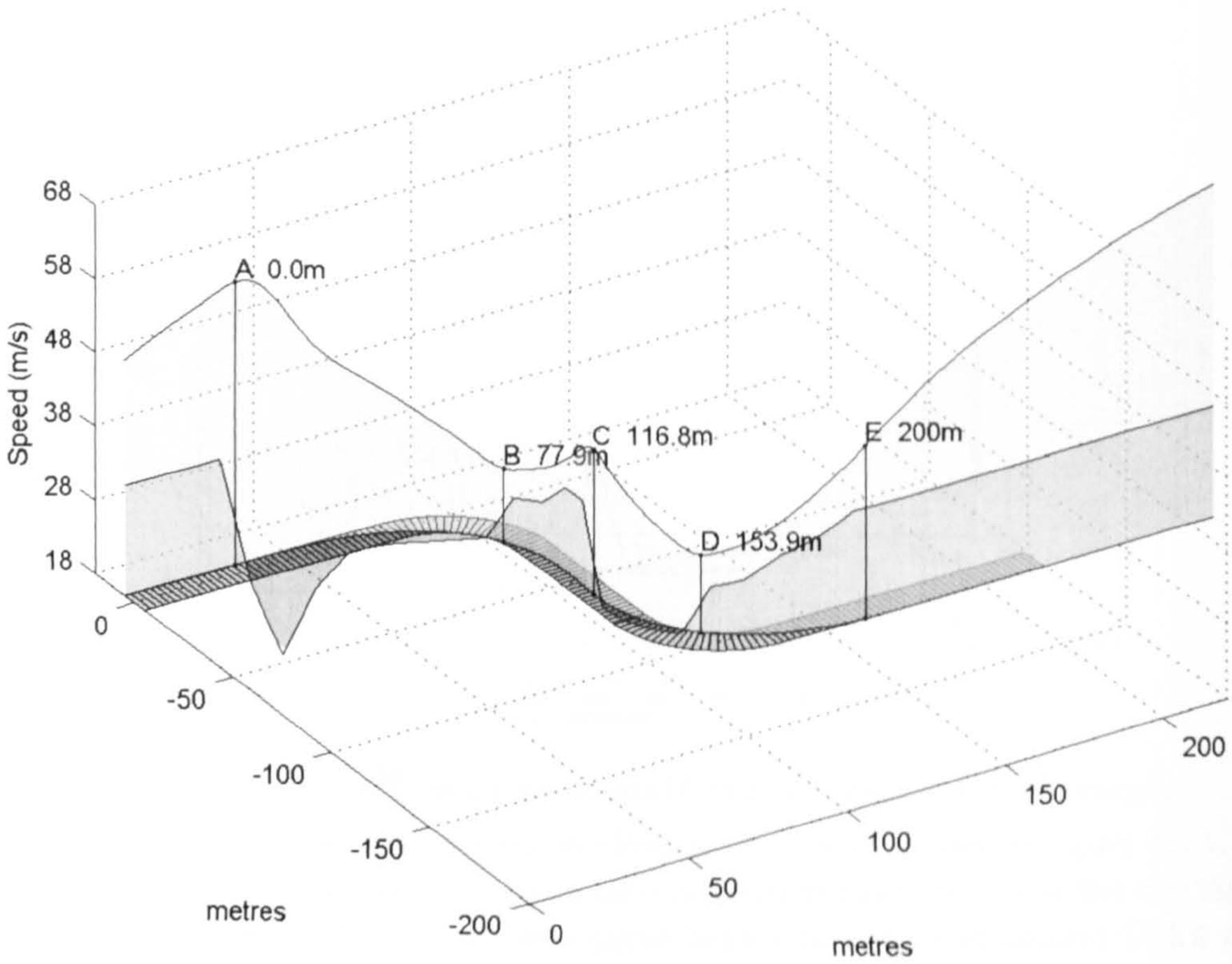


Figure 6.14: Coupled Corner corner 3D view

6.4 Notes on Driven Line and Preview requirements

In this section we discuss the compromises encountered in choosing an optimal driven line and the preview distances required to generate optimal driven lines. We will consider first the case of a corner that is preceded and followed by a long straight such as the 90° righthander. This corner can be considered in isolation. In driving the corner the racing driver is trying to keep as close as possible to the extremes of the acceleration capability of the car (as approximated by the gg-velocity diagram see Appendix D). At the minimum speed point in the manoeuvre the following apply:

- Longitudinal tyre forces are small and the car speed is limited by the path curvature and the lateral capability of the vehicle/tyres.
- The path curvature of the driven line will be close to its maximum value.
- The vehicles longitudinal acceleration changes sign.

The corner can be considered to consist of two phases preceding and following the minimum speed point in the manoeuvre. In choosing the driven line and speed profile the driver is trading off the demands of each phase. The sweep of preview values provides a convenient set of approaches to the corner since we can expect the 'short sighted' approach

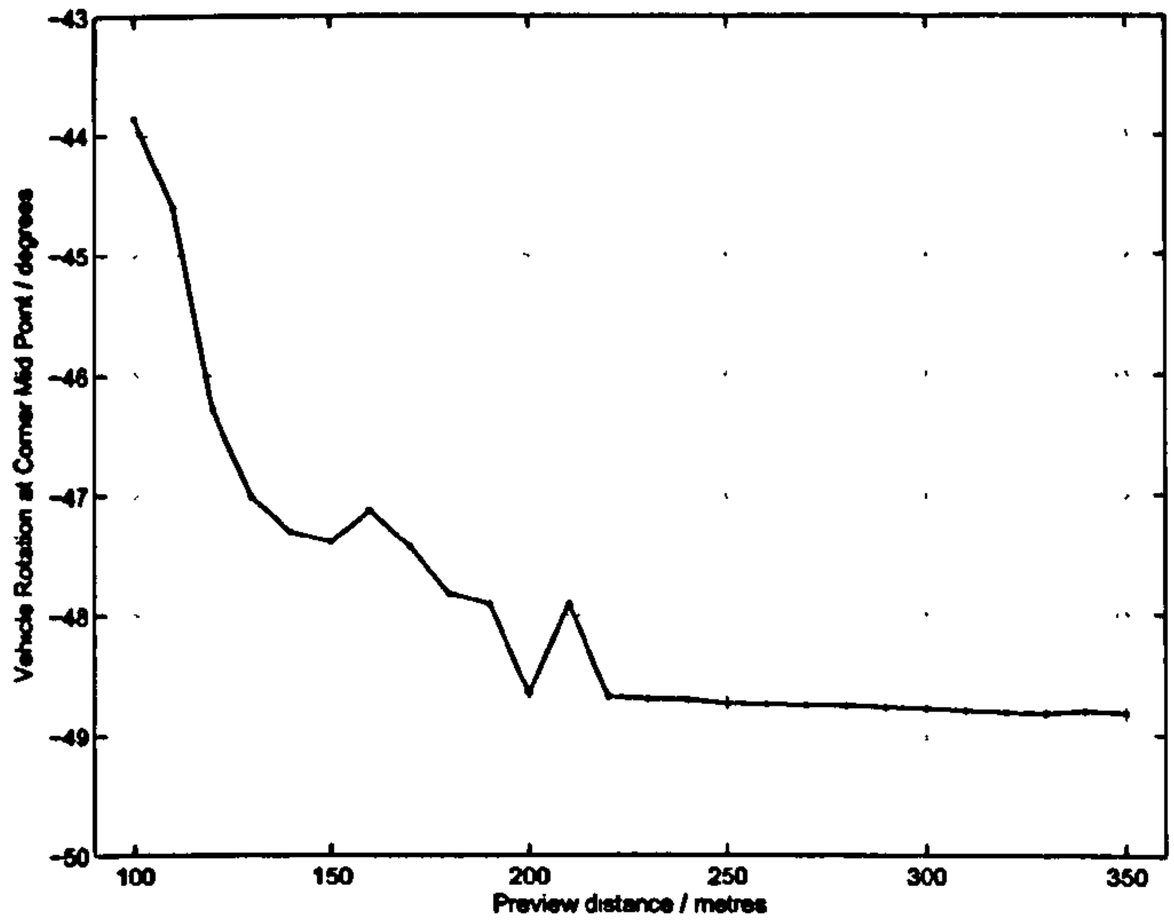


Figure 6.15: Single 90° corner vehicle z axis rotation at Apex (mid-point)

of the short preview to prioritise corner entry over corner exit. The split times shown in figure 6.4 and corresponding apex/exit speeds 6.5 demonstrate this fact. If the car arrives at the mid-point of the corner in less time and with a higher speed, then the corner exit phase takes more time and the final speed is lower. This effect can also be seen in figure 6.15. In the case of the constant radius 90° corner we know that the car must rotate by 90° through the corner. This figure shows that for shorter preview distances there is more rotation left to complete after the midpoint. In the case of this single isolated corner followed by a 200m straigt the minimum manoeuvre time is achieved with a bias toward corner exit speed.

We have shown that in choosing the line and associated speed profile the optimizer must balance the compromise of entry, apex and exit speeds. We have also shown that there exists a preview distance beyond which little reduction in manoeuvre time is obtained. When optimizing a single corner preceded and followed by a long straight, the optimizer must see enough of the following straight to move the compromise in the direction of exit speed. In general, the optimal line through any corner depends on the corners preceding and following. Considering the coupled corners case, we can see that the optimizer needs a preview from the braking point for the first corner through to approximately 50 metres onto the straight following the second corner. A circuit can be considered as consisting of corners and straights. If two right-handed corners are separated by a sufficient length of straight it seems reasonable to assume that they can be optimized independently. There is a point on the exit of the first corner that lies at the minimum sufficient preview distance beyond the braking point. If the braking point for the second corner is further up the track than this, the corners can be considered de-coupled. If this point is beyond the optimal braking point for the second corner then the optimizer needs to see through to the exit of the second corner when optimizing the entry of the first. The analysis is less straightforward for the right-hander followed by left-hander since the vehicle will need to move across the track in between the corners. It seems reasonable to assume that the connecting straight would need to be longer to de-couple the corners under these circumstances.

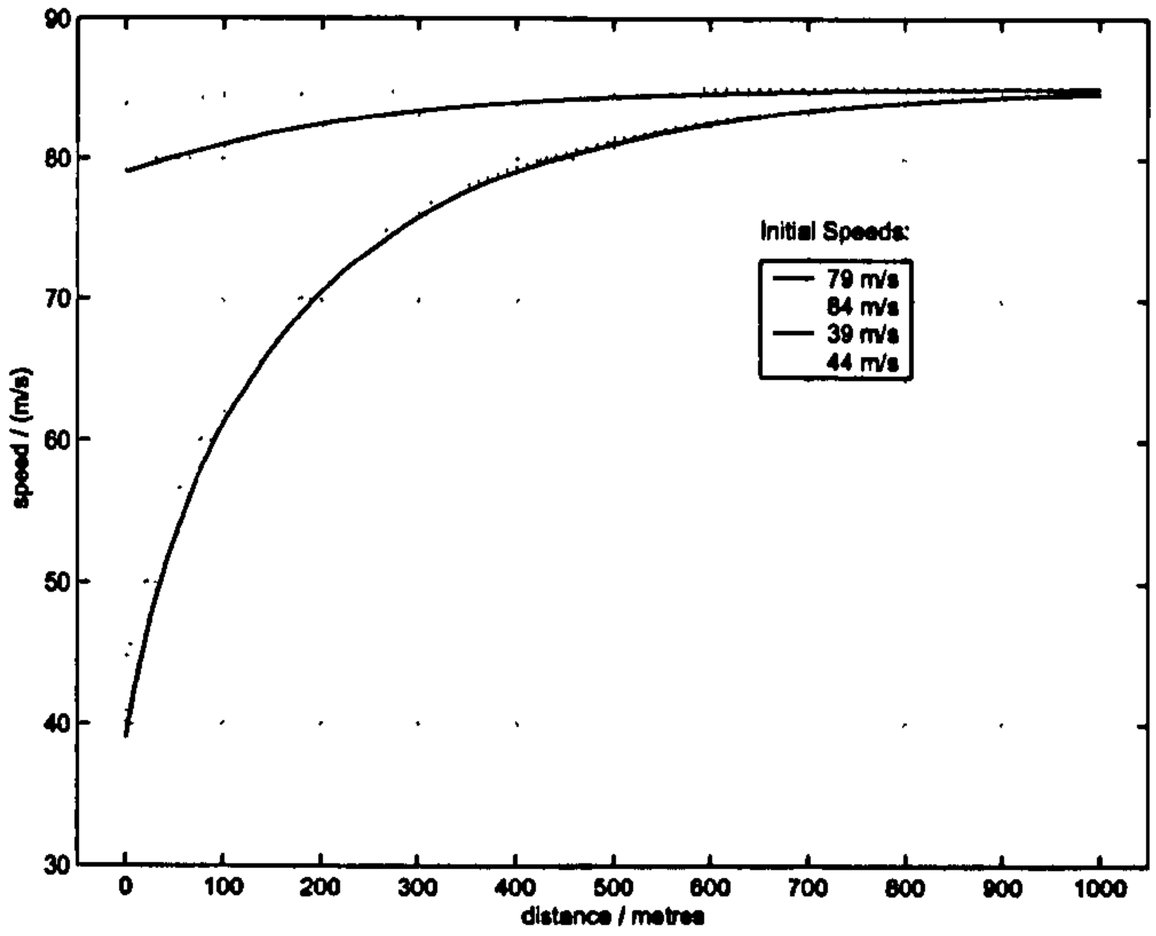


Figure 6.16: Speed along 1km straight

6.4.1 Exit Speed onto a long straight

A typical motor racing circuit will have a main straight with a length in the region of 1km in length. We have shown that an increase in preview distance leads to an increase in exit speed. It is worthwhile to consider how much the additional exit speed is worth in terms of lap time reduction. In this section we consider the effect of initial vehicle speed on the time to traverse a 1km straight. Figures 6.16 shows some example speed profiles for three initial speeds onto the 1km straight. Figure 6.17 shows the time to traverse a 1km straight for initial speeds in the range 35 to 85 m/s. This plot indicates a time reduction of around 35ms for each 1 m/s increase in initial speed. If we consider the final speed against preview distance for the coupled corner manoeuvre shown in figure 6.13, the exit speed benefit from increasing preview from 220 to 300m is around 0.1 m/s. If this manoeuvre was followed by a 1km straight the resulting reduction in maneovre time would be only 3.5ms.

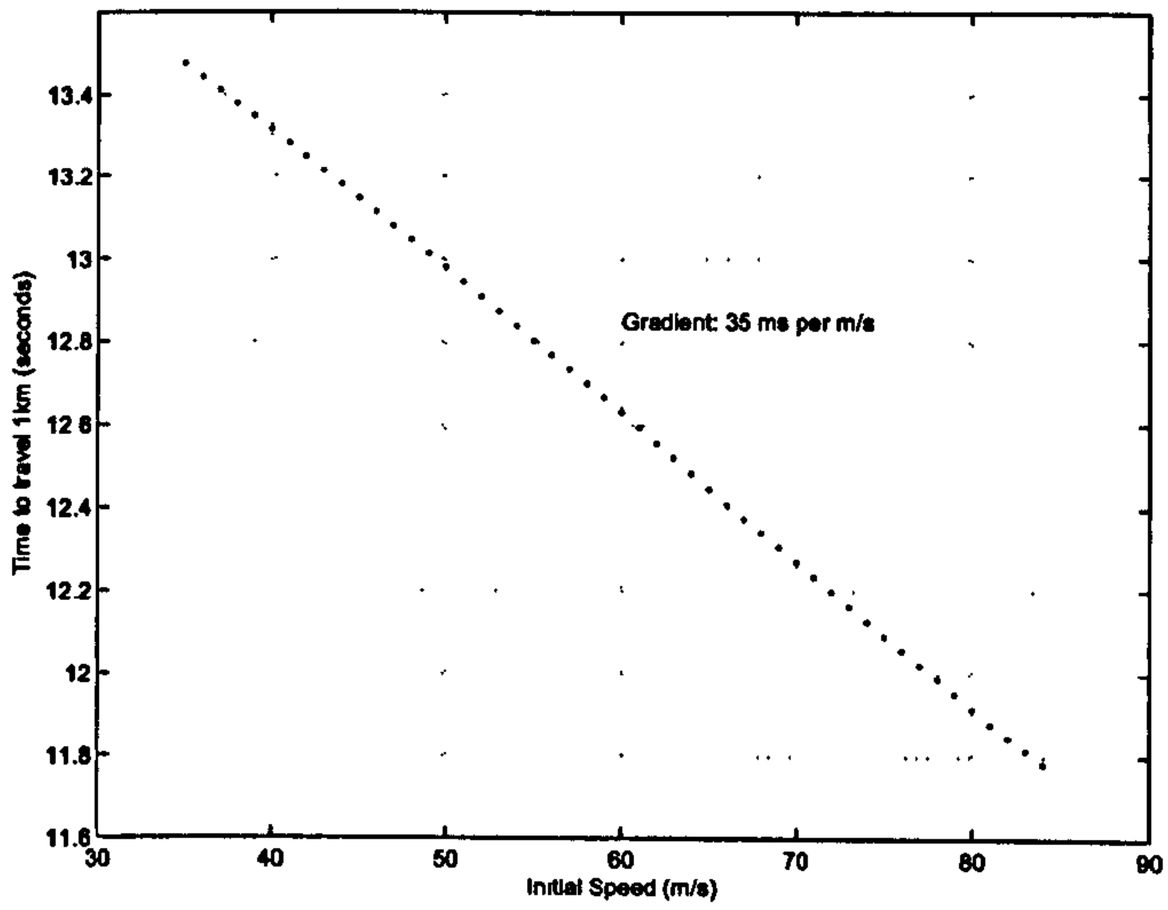


Figure 6.17: Manoeuvre time effect of initial speed

6.5 A full circuit case: Jerez

In this section we apply the finite horizon procedure to a typical motor racing circuit, Jerez. A preview distance of 250m was used. The driven line, speed and controls are shown in figures 6.18 and 6.19.

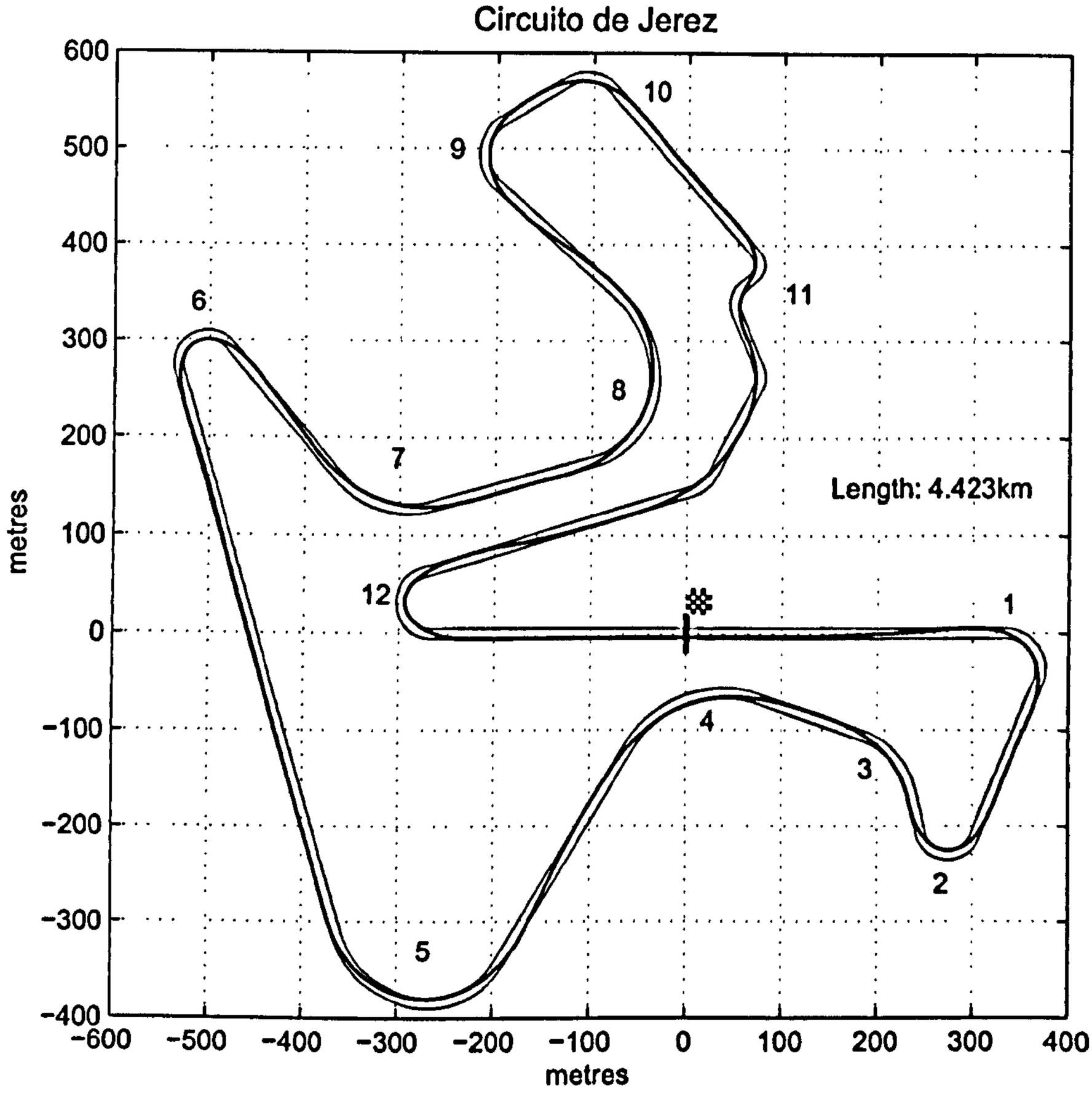


Figure 6.18: Jercz Circuit

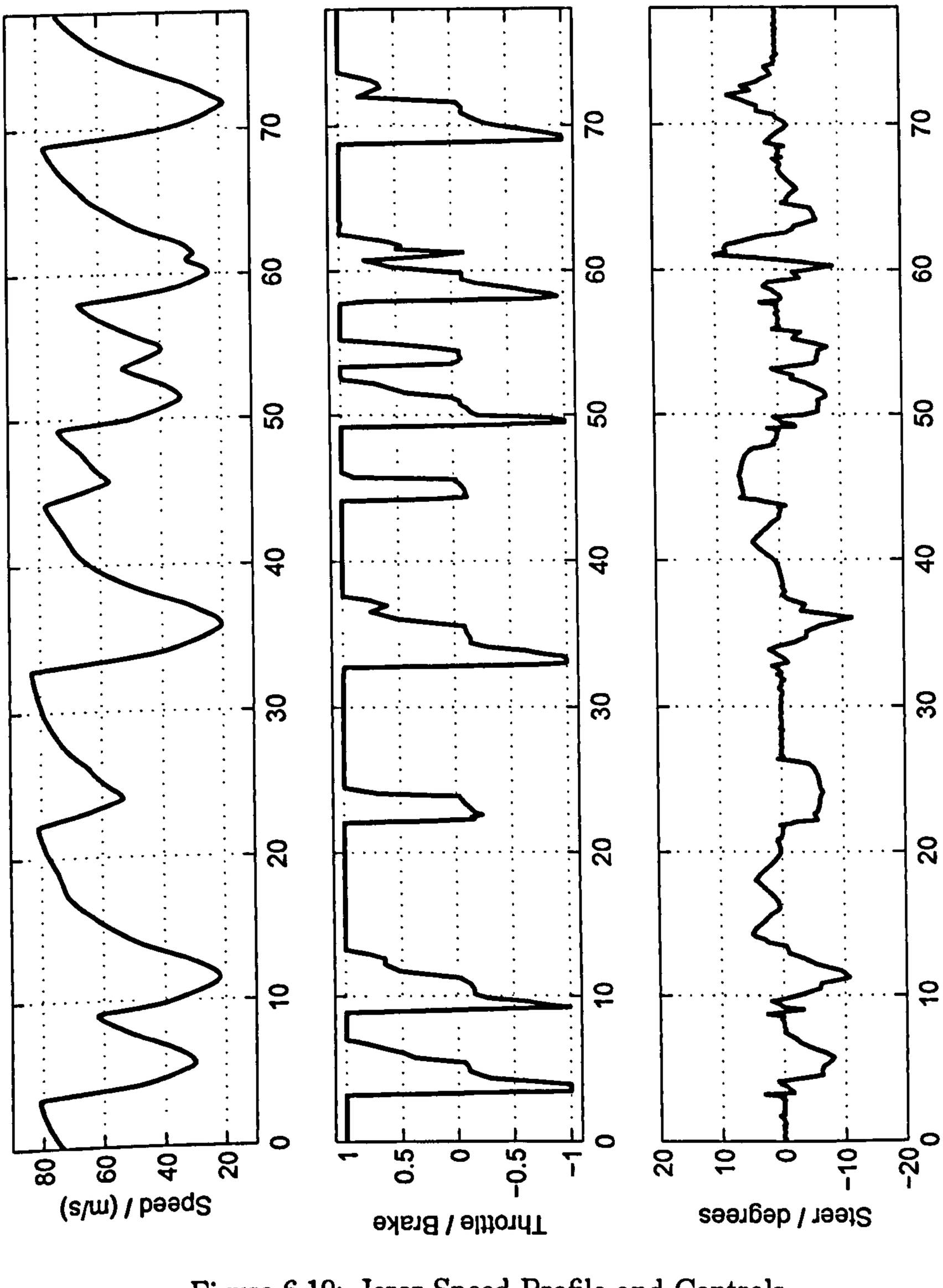


Figure 6.19: Jerez Speed Profile and Controls

Chapter 7

Results using a Sprung Vehicle Model

> The history of damper development is virtually as old as that of the car itself
>
> John C. Dixon

7.1 Summary

In this chapter we demonstrate that the MTM method is capable of operation with a more complex multi-body vehicle model featuring a suspension with spring and damper elements. We consider the performance and stability of the vehicle running over smooth surfaces and over a simulated kerb.

7.2 Sprung vehicle Model

The vehicle model is developed using the mutlibody code Autosim *. Autosim uses the method of generalised speeds [14]. Through symbolic manipulation of the model specification this method eliminates the constraint terms allowing the model to be represented as an ordinary differential equation (ODE). Autosim produces very efficient code in the 'C' or FORTRAN language to compute the model state derivatives. The model can hence be solved using a low complexity ODE solver. A second order Runge-Kutta solver with a time step of 0.5 ms is used in this work. Unfortunately, Autosim is no longer commercially available. The autosim model source listing and vehicle axis system are given in Appendix A). The driveline and aerodynamic model are shared with the QSS vehicle model and are also described in Apendix A.

The model consists of 17 bodies, a chassis, 4 pairs of suspension links, 4 wheel carriers (hubs or uprights) and four wheels. The chassis, suspension links and uprights are modelled

*©Mechanical Simulation Corporation, Ann Arbor, Michigan, USA (www.carsim.com)

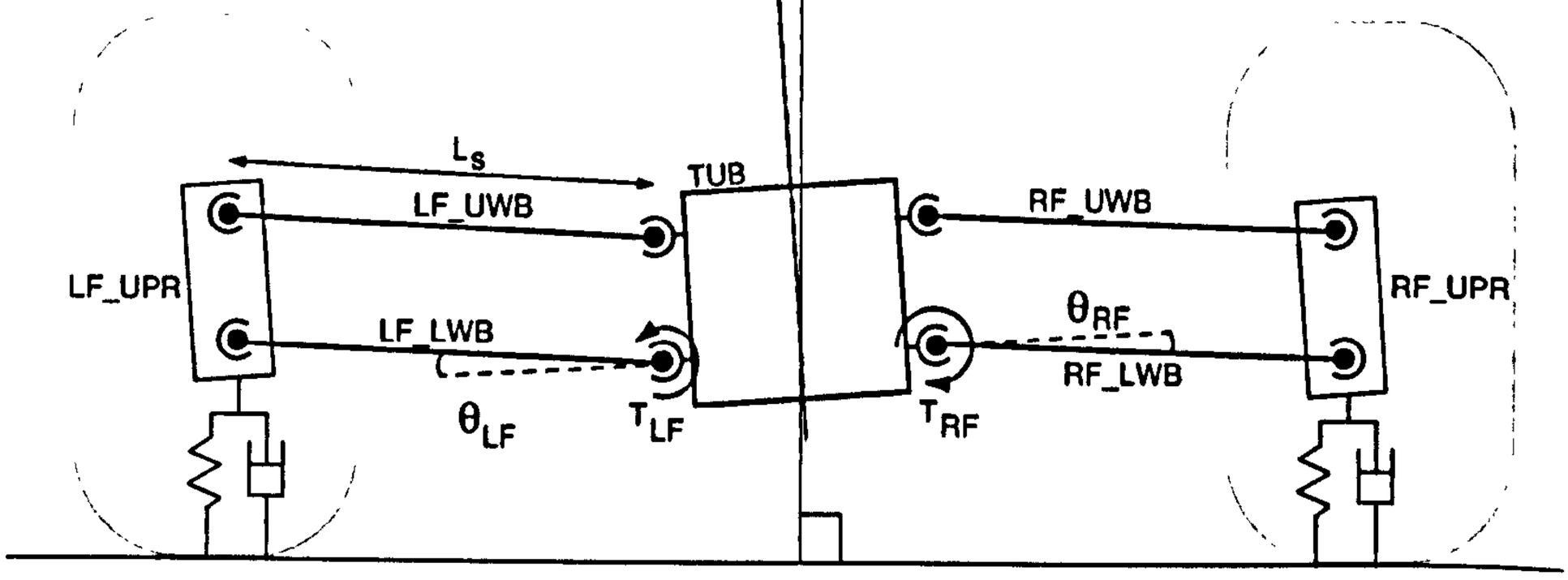


Figure 7.1: Sprung Car Suspension

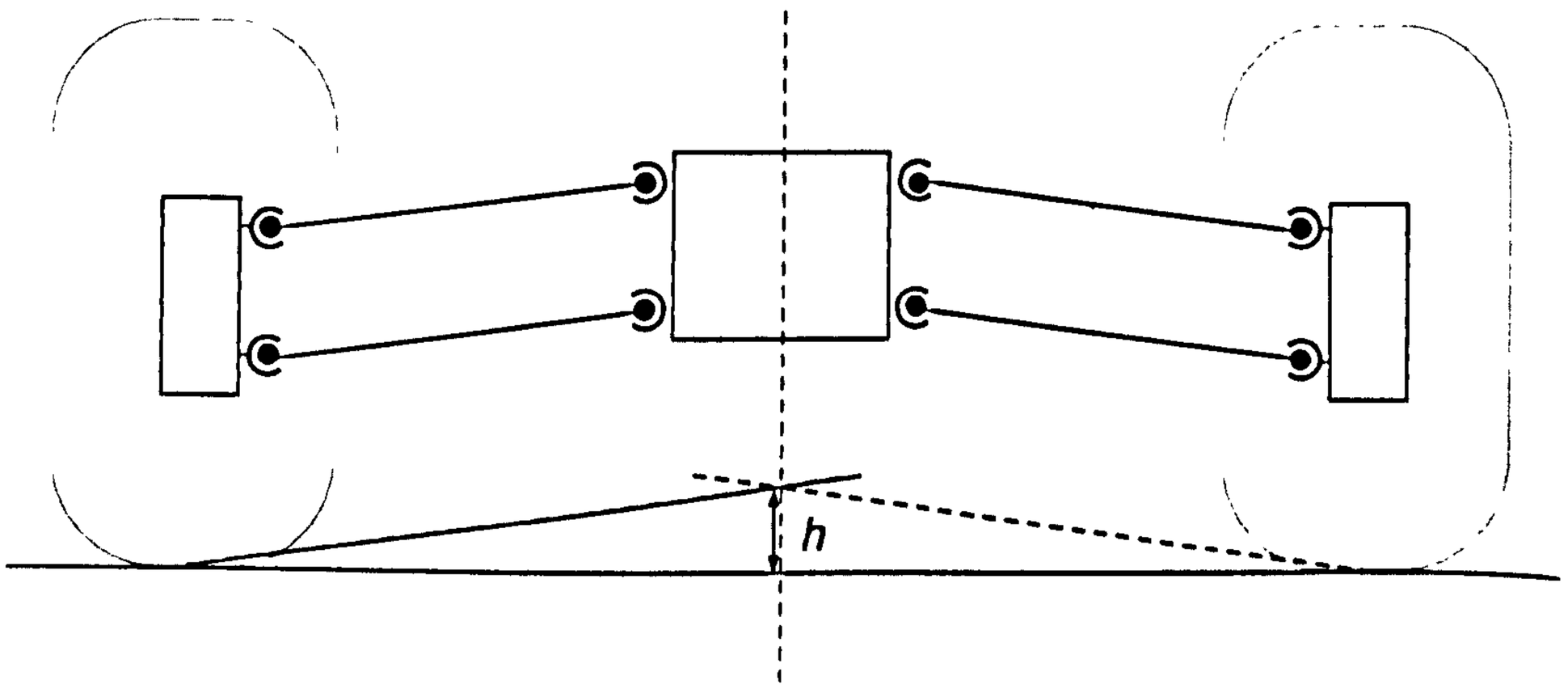


Figure 7.2: Sprung Car Roll Centres

in autosim. The suspension links have zero mass. The wheels are handled outside of autosim in separate code. Each has a single degree of rotational freedom relative to the hub. Gyroscopic forces due to the rotating wheels are neglected. The front and rear suspensions of the car are modelled as shown in figure 7.1. Each of the unsprung masses (uprights) is connected to the sprung mass by two massless rigid 'suspension' links. Each link is connected to the chassis and one unsprung mass by a revolute joint. Each link is constrained to rotate about a line parallel to the vehicle x axis passing through the point of connection between the link and the chassis. The link is constrained to lie in the vehicle z, y plane passing through this point. The joints between the links and unsprung masses are similar with the each unsprung mass closing the kinematic loop. Each corner of the car hence has a single degree of freedom. We choose the angle formed between the lower

suspension link and the vehicle y axis to describe this $\theta_{LF}, \theta_{RF}, \theta_{LR}, \theta_{RR}$. This 'parallel arm' suspension model results in roll centre heights [17] as shown in figure 7.2. The front and rear roll centre heights are given by h_F and h_R respectively.

An increase in z height of an unsprung mass relative to the sprung mass is termed 'bump'. The opposite is termed 'droop'. In order to have similar sign conventions for bump and droop on each side of the car we define the clockwise as positive on the left and anticlockwise as positive on the right. The suspension angle is zero when the link is parallel to the vehicle y axis. Motion of the chassis along its z axis relative to both left and right suspension elements is termed 'heave'. We define the suspension deflection angle corresponding to heave as the sum of the two lower suspension arm angles:

$$\theta_{heave} = \theta_L + \theta_R \tag{7.1}$$

Rotation of the chassis about its x axis is termed roll. We define the suspension deflection angle corresponding to roll as the difference between the two lower suspension angles.

$$\theta_{roll} = \theta_R - \theta_L \tag{7.2}$$

We define θ_{heave_F},θ_{heave_R},θ_{roll_F} and θ_{roll_R} as the heave and roll deflection angles at the front and rear axles. Note that the more correct definition of roll is that which takes place when the vehicle is cornering. In this case, the vehicle rotates about its roll axis (the line passing through the roll centres). In our axis system this is a combination of rotation about the vehicle x axis and translation. The model applies the suspension forces as torques acting between the chassis and lower suspension arms. The computation of these torques is discussed below.

7.2.1 Suspension

The job of the suspension is as follows:

1. To maintain the optimal load on the tyre, minimising load fluctuations due to bumps in the road surface

2. To maintain the height and attitude of the chassis relative to the ground as closely as possible to ideal values specified by aerodynamic requirements.

3. To provide a desired kinematic position and orientation of each tyre relative to the ground and to the other tyres (toe, camber).

4. To position the roll centres and anti-dive / anti-squat to provide the desired load transfer through the suspension links

5. To provide self-aligning forces for the steered wheels and to support the desired position and orientation of the steered wheels (as in item 3) through the range of suspension travel

6. To allow lateral load transfer forces to be distributed between the front and rear axle.

In this work we restrict our scope mainly to item 1 and by default to the roll centre heights (item 4). The simple aerodynamic model we use does not include ride height effects so item 2 is not considered. In general however, the suspension stiffnesses are capable of supporting aerodynamic loads and are representative of those used in practice (at the lower end of a rising rate suspension at least). Items 3 and 5 are not considered. The model suspension generates a camber angle due to the rotation of the unsprung masses as the chassis rolls in cornering, however this camber angle is not used in the tyre force calculation. Hence all tyre forces are generated assuming the tyre has zero camber. The model can readily be extended to include camber and toe [17] if required. Item 6 is a fundamental tuning tool used by engineers to balance the yaw moment of the car in cornering. It is discussed further below.

7.2.2 Suspension schemes

As previously discussed, the suspension forces are modelled as torques between the chassis and lower suspension link. Here we discuss how the values of these forces are calculated. A typical racing car suspension system using one or more of the following:

- 'Heave' elements
- 'Roll' elements
- 'Side' elements

A heave element responds to the suspension deflection that is common to both sides on the same axle. The 'heave' terms act equally on both sides of the axle and resist vertical motion of the chassis relative to both wheels. The heave forces hold the car off the ground, supporting the chassis mass and aerodynamic load. Since in our model the heave forces are independent for each axle, our heave elements also control the pitching motion of the chassis. A roll element resists the rolling motion of the chassis. It applies torque so as to try and equalise the angles of the suspension links on the left and right sides. It applies a force of equal magnitude but opposite sign to each side of the axle. A side suspension element applies forces to only one corner of the car. It contributes to both the heave and roll effects. For simplicity, we do not consider side elements in this work.

In the equations below we define the torque acting between the chassis and each lower suspension link of the car using a combintation of heave and roll terms. Each of these torques has a 'spring' term that is proportional to suspension angle and a 'damping' term that is proportional to the suspension velocity. The suspension link length is given by L_s.

$$T_{heave} = L_s K_{heave}((\theta_R - \theta_0) + (\theta_L - \theta_0)) - L_s C_{heave}(\dot{\theta_R} + \dot{\theta_L}) \tag{7.3}$$

$$T_{roll} = L_s K_{roll}((\theta_R - \theta_0) - (\theta_L - \theta_0)) - L_s C_{roll}(\dot{\theta_R} - \dot{\theta_L}) \tag{7.4}$$

$$T_R = T_{heave} + T_{roll} \tag{7.5}$$

$$T_L = T_{heave} - T_{roll} \tag{7.6}$$

Hence the suspension configuration on each axle is defined by the five parameters shown in table 7.2.2.

Symbol	Description
θ_0	Suspension angle for zero force
C_{heave}	heave damping rate (Ns/m or Ns/mm)
C_{roll}	Roll damping rate (Ns/m or Ns/mm)
K_{heave}	heave rate (N/m or N/mm)
K_{roll}	Roll rate (N/m or N/mm)

Table 7.1: Suspension Parameters per axle

Symbol	Description	Value
K_{tyre}	Tyre Spring Rate	270 N/mm
C_{tyre}	Tyre Damping Rate	1 Ns/mm

Table 7.2: Tyre vertical stiffness and damping

The rates are specified in units of N/m or Ns/m and refered as angular rates by multiplying with the suspension link length L and assuming suspension deflection angles are small.

We separate the heave and roll elements in this manner to allow their effect to be studied independently. Independent heave and roll rates can be readily realised in practical race car suspensions. Independent heave damping is also easy to implement. A pure roll damping term would be harder to implement and to the authors knowledge such a system is not in widespread use. It is also possible to envisage suspension elements that act only on vehicle pitch through suitable coupling of the front and rear axles.

7.2.2.1 Wheel Rates and Contact Patch Rates

As defined above, the spring and damping rates describe forces generated between the 'unsprung' masses and the chassis. We will refer to these as 'wheel' rates. The unsprung masses are supported by the tyre. We model the vertical dynamics of the tyre as a spring and damper element in parallel with coefficients K_{tyre} and C_{tyre} respectively. This leads to an overall tyre and suspension rate consisting of the suspension and tyre springs/dampers in series. We refer to this as the 'true' rate or 'contact patch' rate. For example, equation 7.7 defines the roll stiffness at the contact patch.

$$K_{roll_{cp}} = \frac{1}{\frac{1}{K_{tyre}} + \frac{1}{K_{roll}}} \tag{7.7}$$

As wheel rates increase, the tyre becomes a more significant factor in the contact patch rate. For example, as overall roll stiffness increases, the true roll moment distribution will approach that of the tyre springs. The values for the stiffness and damping coefficients of both the front and rear tyres is given in table 7.2.

7.2.2.2 Rising Rate Suspension

Suspensions for racecars with large aerodynamic loads typically have spring rates that change with suspension displacement. The suspension links are arranged so that the wheel rate increases as the suspension moves further into bump. This allows lower wheel rates to be used at low speed (low aerodynamic load) whilst still holding the car off the ground at high speeds. We do not consider rising rate suspensions here but it is a simple exercise to add this feature to the model.

7.2.2.3 Roll moment distribution

When a car is cornering, the lateral forces at the tyre contact patches produce a moment about the vehicle centre of gravity. This moment is counteracted by an increase in the vertical forces on the outside tyres and a corresponding decrease on the inside. The total load transferred is determined by the total magnitude of the lateral force and the vehicle centre of gravity height. This load can be transferred by either axle. Since the front and rear axles have independent roll stiffness values, and the proportion carried by each axle is controlled by their relative roll stiffnesses and roll centre heights. Equation 7.8 describes the instantaneous fraction of load transfer (at time t) carried by the front axle. This gives the true roll moment distribution (RMD) including loads transferred by suspension links and with roll rates referred to the contact patch (including tyre spring and damping rate).

$$D_{roll_{cp}}(t) = \frac{|F_{zRF}(t) - F_{zLF}(t)|}{|F_{zRF}(t) + F_{zRR}(t) - F_{zLF}(t) - F_{zLR}(t)|} \tag{7.8}$$

We define the roll stiffness distribution due to the roll rates alone in equation 7.9. This is the ratio of the front roll stiffness to the total roll stiffness.

$$D_{roll_{spring}} = \frac{K_{roll_F}}{K_{roll_F} + K_{roll_R}} \tag{7.9}$$

The coefficient of friction of the tyre (Appendix B) decreases with load. Hence when cornering, the loaded side of the car sees a reduced coefficient of friction. The RMD can be used to reduce the coefficient of friction on either axle. For example, moving the RMD forwards increases rear axle and decreases front axle coefficient of friction hence promoting understeer [17]. This is a primary tool used by trackside engineers to shift the oversteer/understeer 'balance' of a car. Increasing load transfer on the front axle also produces an understeering moment on the chassis as the front tyre lateral forces have a component in the negative vehicle x direction (scrub).

7.2.2.4 Heave Stiffness Distribution

For convenience we also define a heave stiffness ratio in equation 7.10.

$$D_{heave_{spring}} = \frac{K_{heave_F}}{K_{heave_F} + K_{heave_R}} \tag{7.10}$$

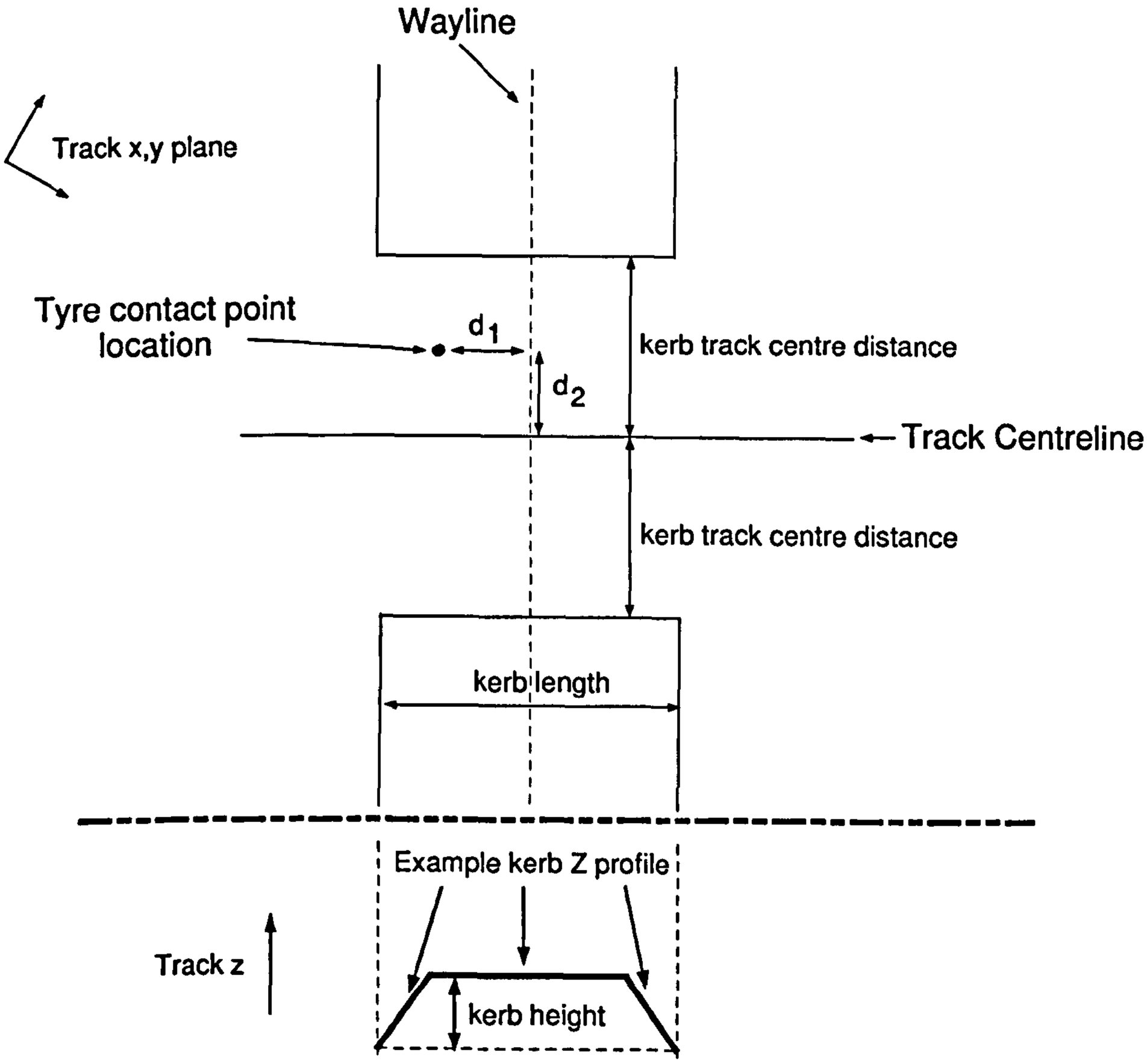


Figure 7.3: Implementation of bumps/kerbs

7.3 Implementation of Bump and Kerb data

All of the results shown in chapter 4 are produced with the assumption that the tyre contact patches are moving on a flat (x, y) plane. In this chapter we add another dimension in order to displace the tyre contact patches in the z direction to allow the simulation of bumps or kerbs in the track. This is achieved by associating a height profile with each wayline (section 2.3) as shown in figure 7.3. We simplify the tyre/road contact to a single point (this is of course quite unrepresentative). At each time step taken by the ODE solver we compute first the x and y locations of the points directly below the centre of each wheel. These coordinates are then transformed into the coordinate system of the nearest waypoint. This gives distances d_1 and d_2 which become the arguments to a function used to compute the height of the road at that point.

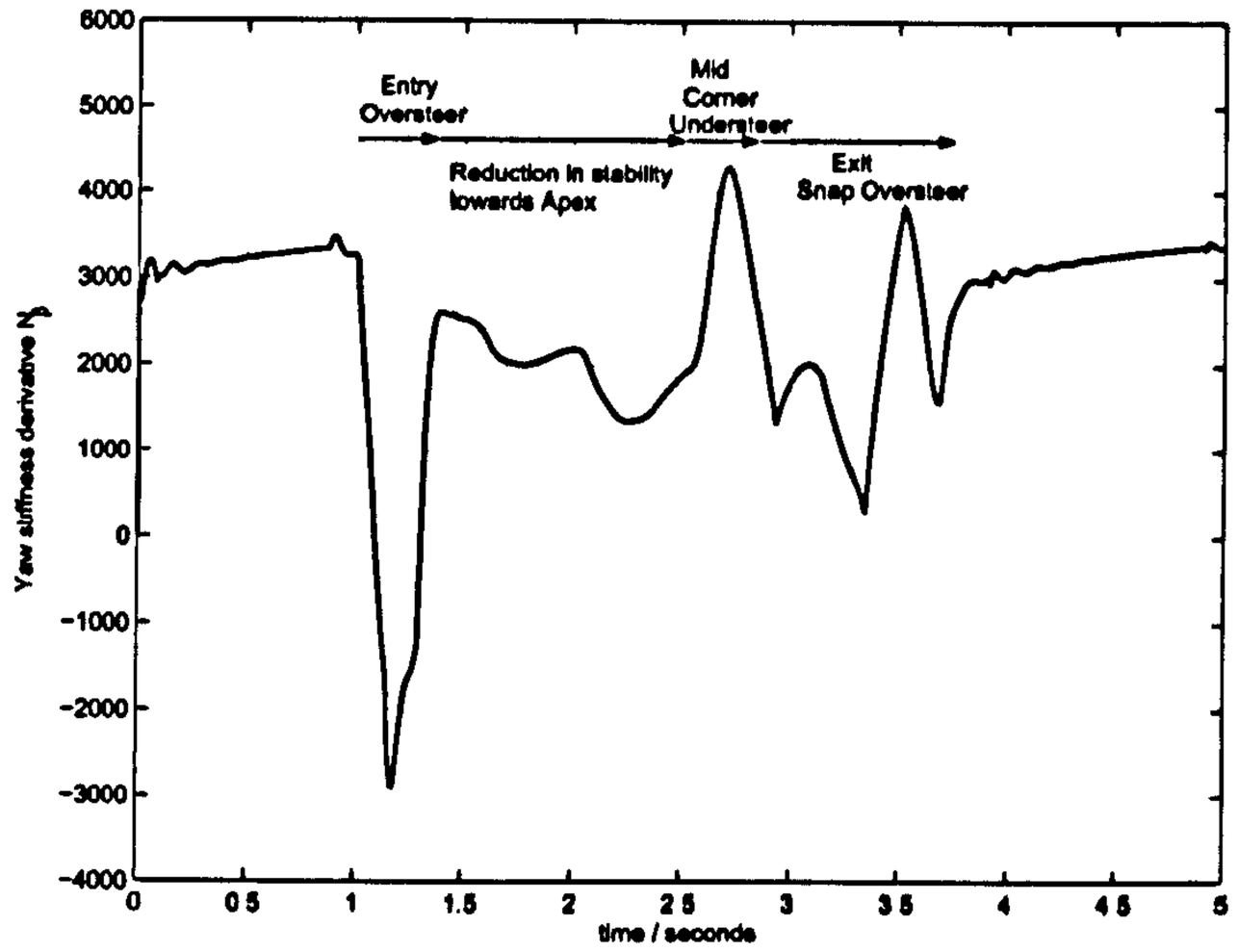


Figure 7.4: Yaw Stiffness Derivative

7.4 Vehicle stability through the corner phases

We begin the study of the sprung vehicle behaviour with an analysis of the yaw stiffness and damping derivatives for the single 90° corner maoeuvre. The distance method is used with a control spacing of 8m. The vehicle is parameterised as shown in table 7.3. The yaw stiffness and yaw damping derivatives are shown in figures 7.4 and 7.5. These plots nicely sum up the typical behaviour of a formula 1 car when cornering. The car is initially unstable at brake-and-turn-in, stability is then restored after initial heavy braking is complete. Stability is then reduced in later entry as the brake torque is reduced. When the power is applied, there is initially excessive stability / understeer followed by a swift reduction in stability / snap oversteer on exit.

Symbol	Description
K_{heave_F}	80 N/mm
K_{heave_R}	40 N/mm
C_{heave_F}	2 Ns/mm
C_{heave_R}	1 Ns/mm
K_{roll_F}	20 N/mm
K_{roll_R}	10 N/mm
C_{roll_F}	1 Ns/mm
C_{roll_R}	0.5 Ns/mm

Table 7.3: Basic Sprung Vehicle Setup

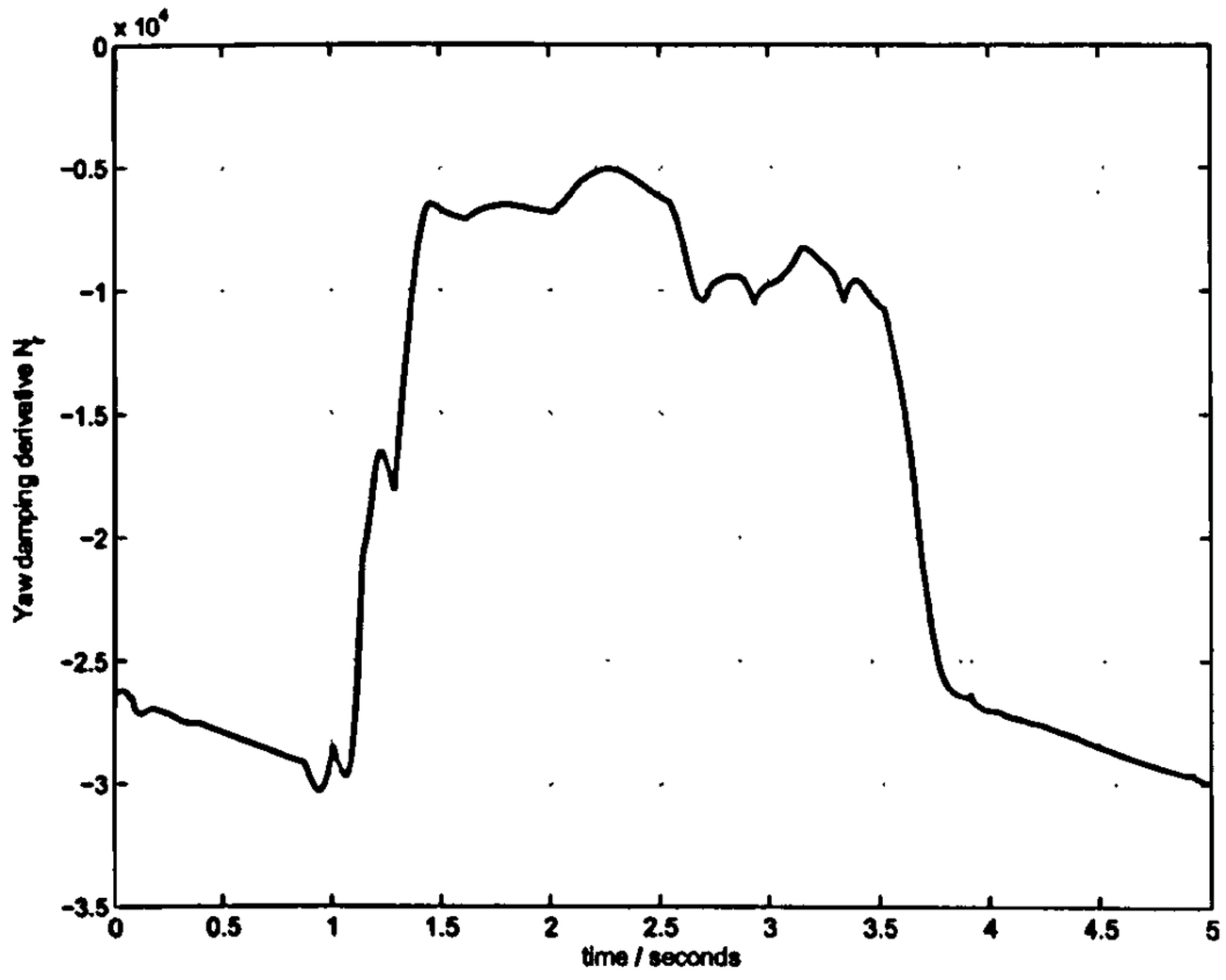


Figure 7.5: Yaw Damping Derivative

7.5 Roll damping and transient stability

In this section we use an exagerated example to show the impact of roll damping on vehicle stability during transient turn entry and exit. During the transient phase of turn entry the chassis has roll velocity resulting in suspension forces from the roll dampers. The roll moment distribution is hence influenced by the relative magnitudes of the front and rear roll damping terms during transient turn entry. The vehicle setup is given in table 7.4. We consider two cases one with forward biased roll damping and on with rearward. In the forward roll damping case $C_{roll_F} = 2$ Ns/mm and $C_{roll_R} = 1$Ns/mm. In the rearward case the values are reversed with $C_{roll_F} = 1$ Ns/mm and $C_{roll_R} = 2$ Ns/mm. Figures 7.6 to 7.11

Term	Value
K_{heave_F}	60 N/mm
K_{heave_R}	30 N/mm
C_{heave_F}	2 Ns/mm
C_{heave_R}	1 Ns/mm
K_{roll_F}	20 N/mm
K_{roll_R}	9.582 N/mm

Table 7.4: Sprung Vehicle Setup for transient roll damping test

show the optimized controls, speed and timeline, yaw stiffness and damping, roll spring and damper forces, D_{roll} and roll angle and tyre efficiencies for the two cases. Figure 7.9 shows the expected change in distribution of damper forces. The rearwards case results in the following:

- more rearward D_{roll} during roll transients

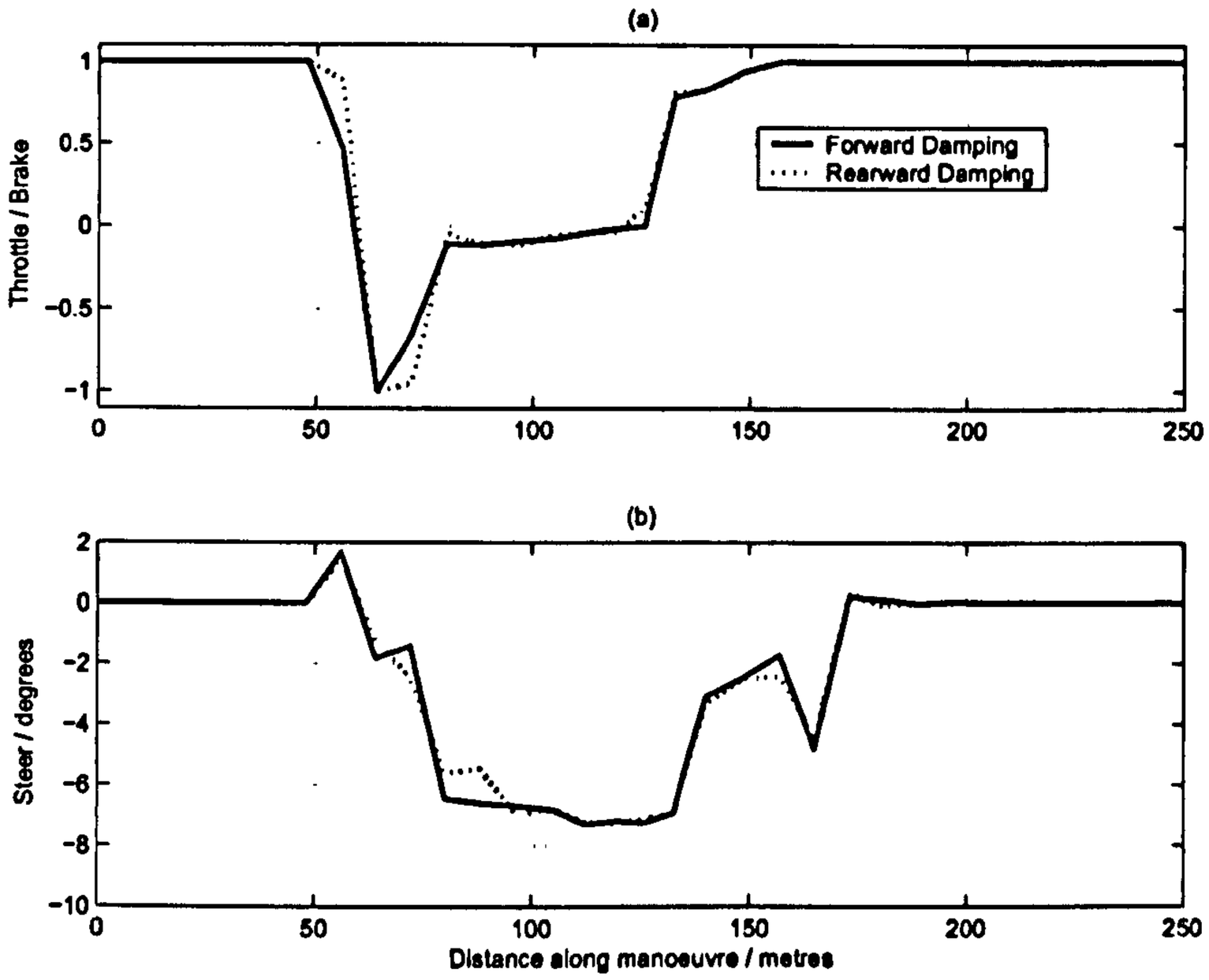


Figure 7.6: Vehicle Controls

- reduced N_{beta} and N_r during corner entry
- The tyre efficiency diagrams show the increased use of the rear tyres tyres in corner entry.

The tyre efficiency diagrams show that in the forwards case the car is front tyre limited throughout the manoeuvre. In the rearwards case, the rear tyres are worked harder in corner entry and the resulting reduction in stability allows the car to rotate onto the corner more readily. This result is line with several other results in this thesis where reduced stability results in increased speed during transient manoeuvering. It is worth noting again that when setting-up a car for a human driver, entry stability is at a premium and its reduction is not a good thing.

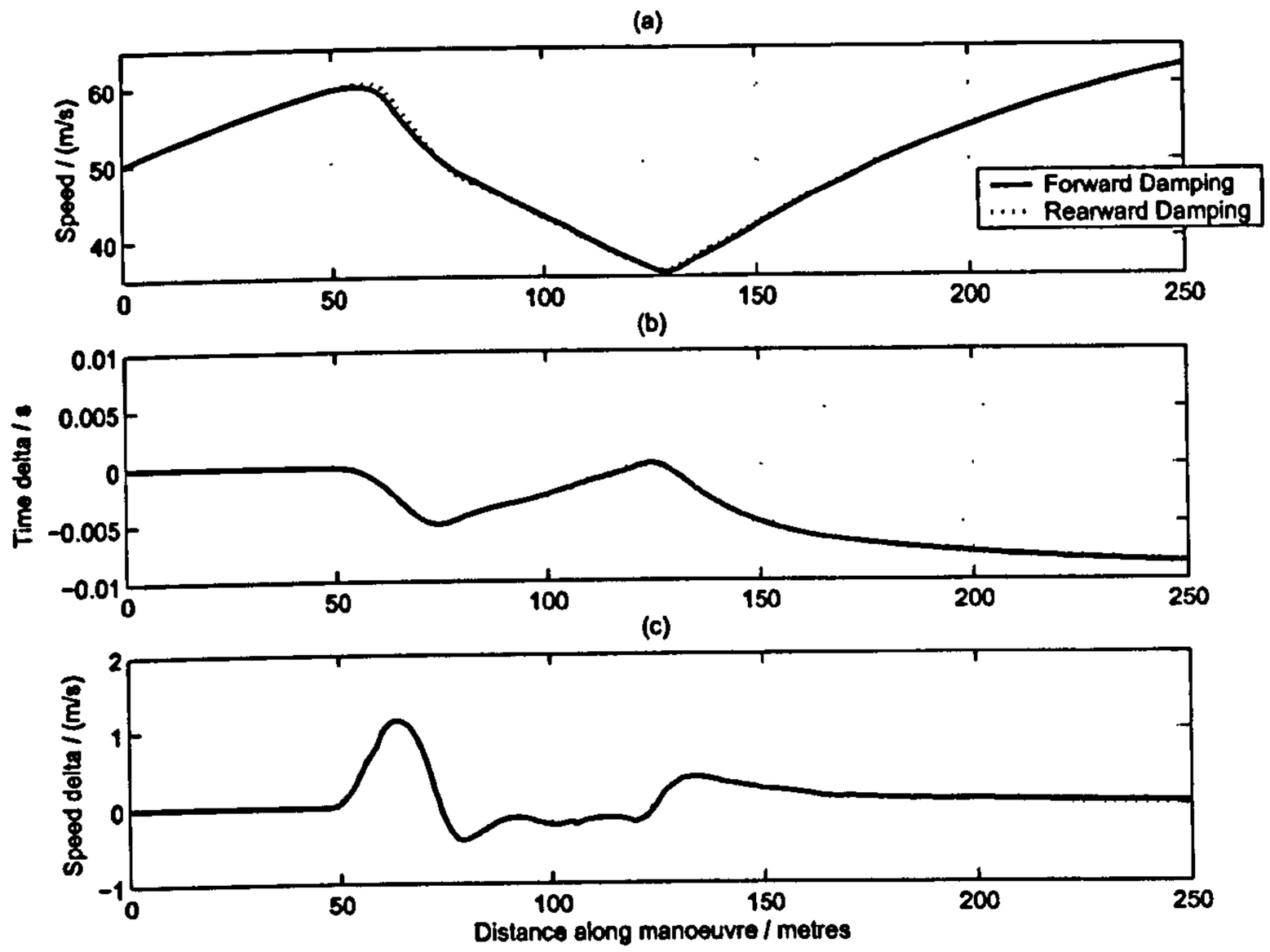


Figure 7.7: Speed and Timeline

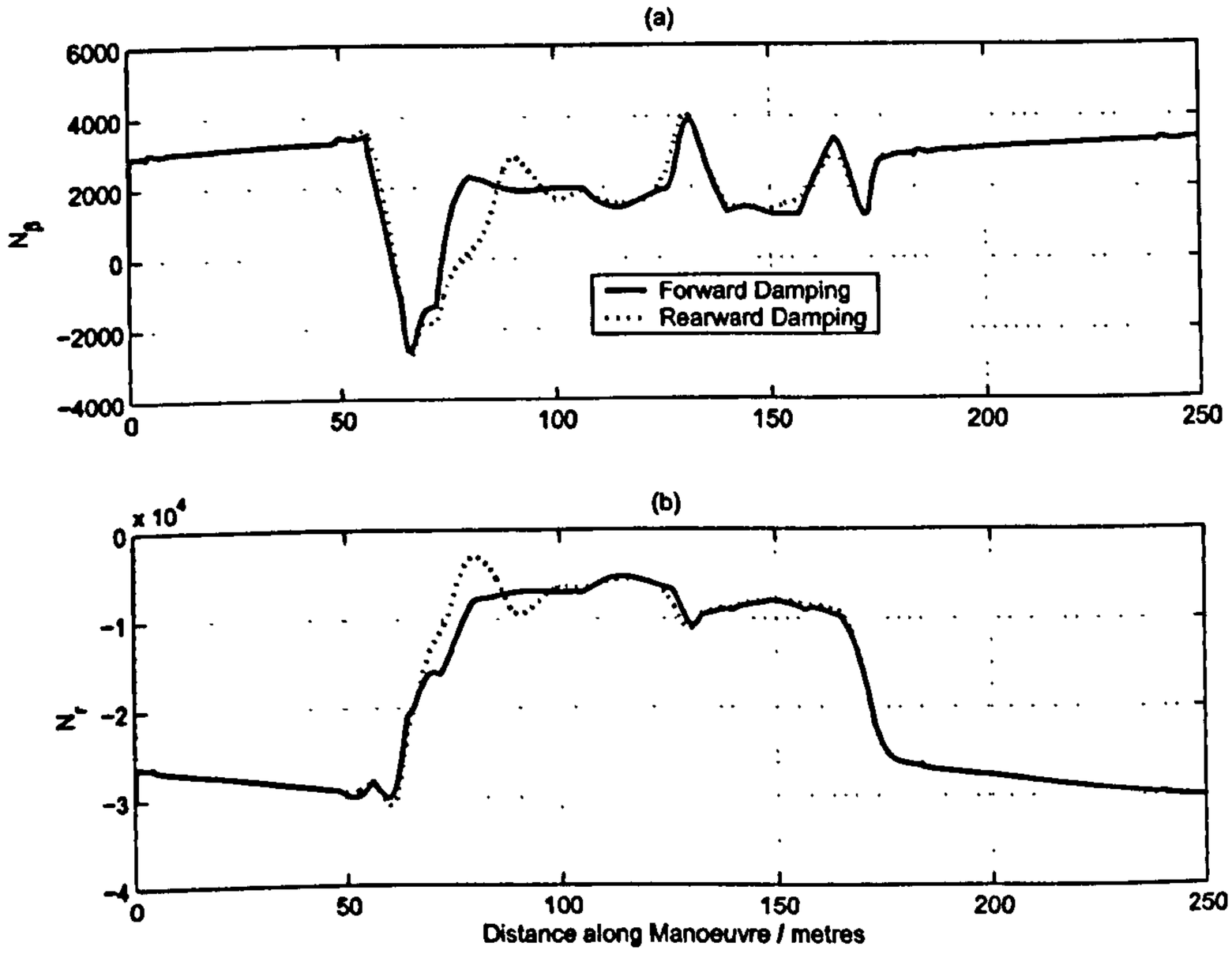


Figure 7.8: Yaw Stiffness and Yaw Damping Derivatives

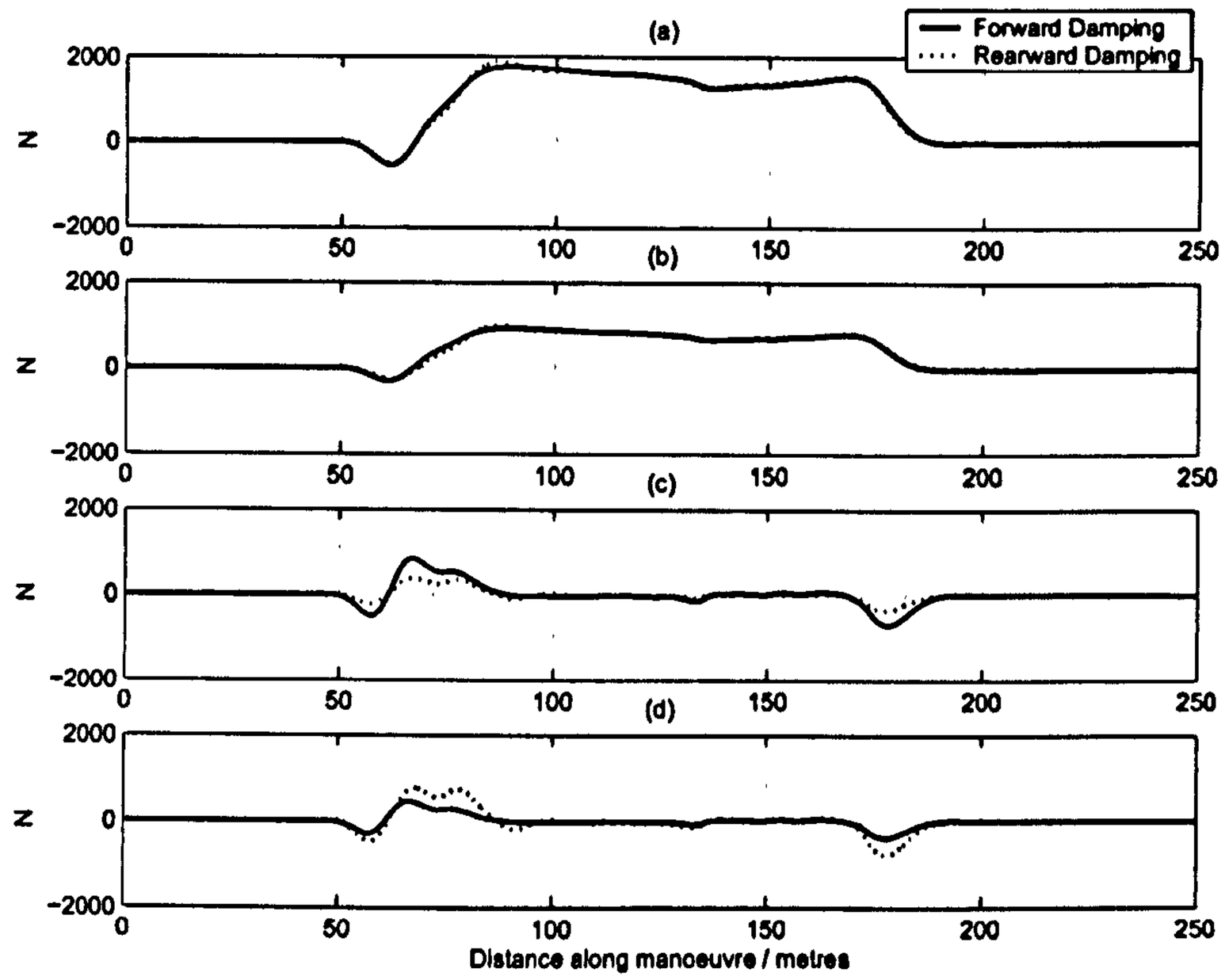


Figure 7.9: Suspension Roll Forces (a) Front Roll 'Spring' force (b) Rear Roll 'Spring' force (c) Front Roll damping force (d) Rear Roll damping force

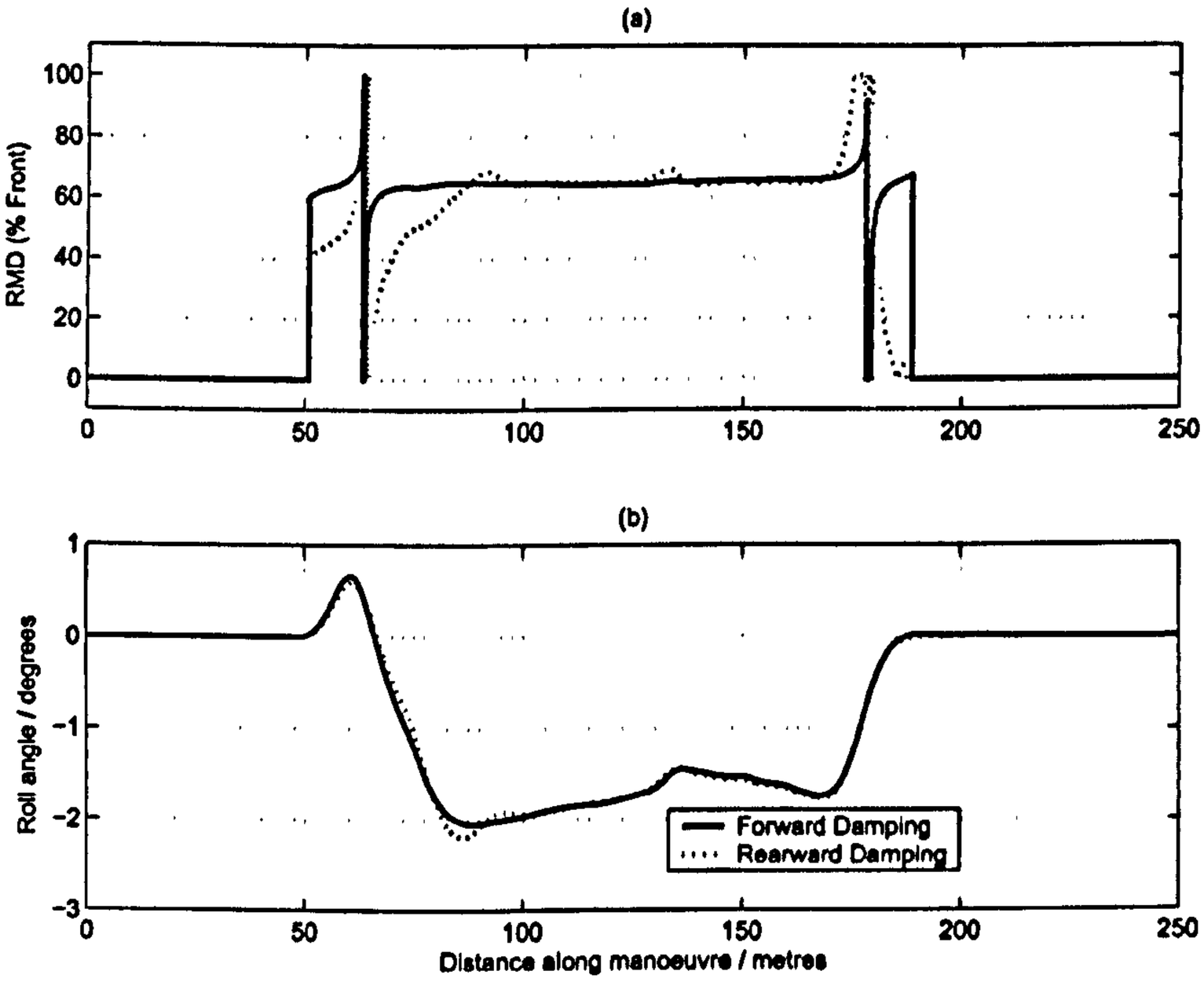


Figure 7.10: (a) Roll moment distribution (b) Roll angle

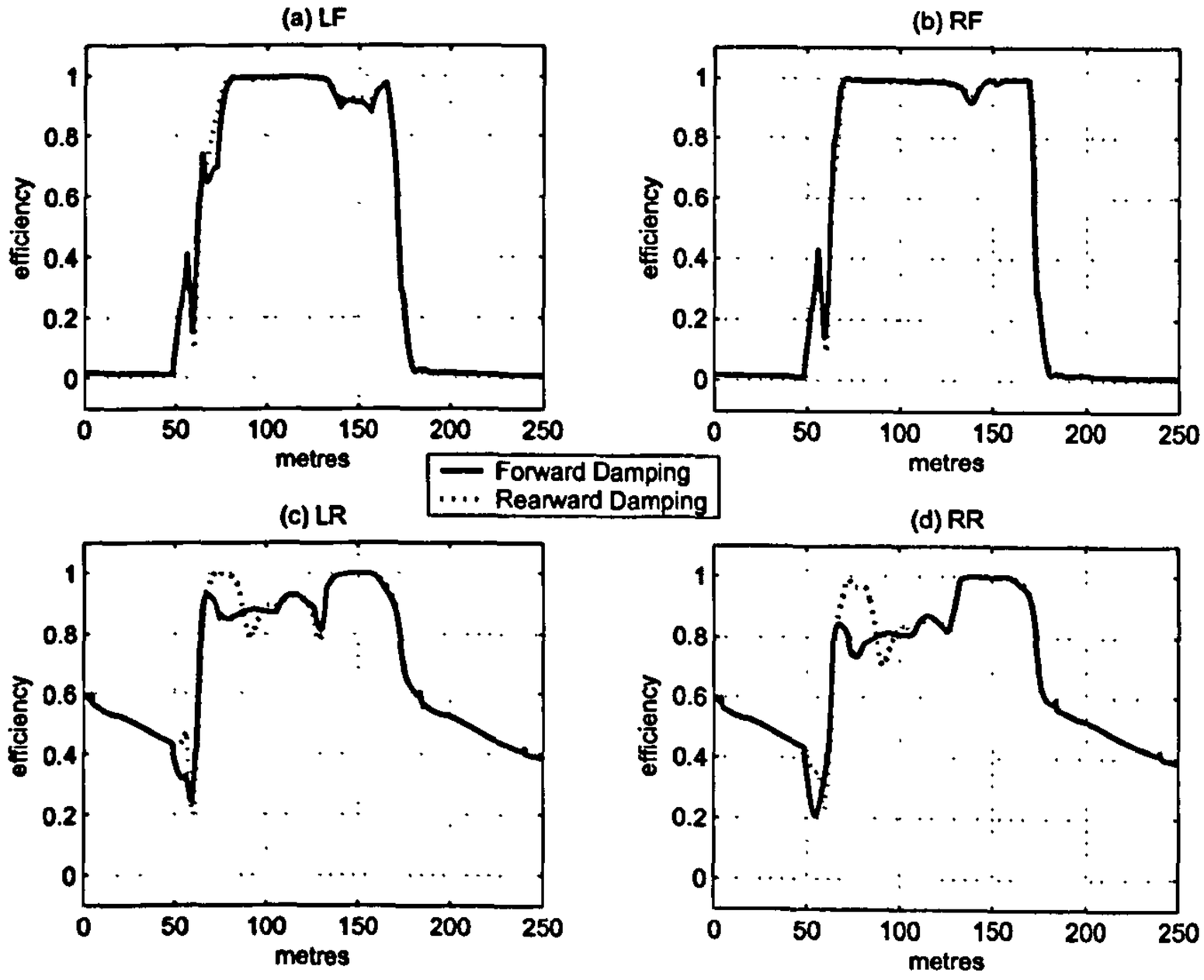


Figure 7.11: Tyre efficiencies

7.6 Kerbing Analysis of Roll and Heave Rates

In this section we examine the effect of roll and heave spring/damping rates on manoeuvre time for the single 90° corner with the addition of a kerb at the Apex of the corner. The profile of each 'bump' in the kerb is shown in figure 7.12. It has a cosine profile with a period of 2 metres. The kerb consists of 6 'bumps' as shown in figure 7.13. In previous chapters we have constrained the minimum load on any wheel to some value greater than zero, forcing the MTM method to keep all four tyres in contact with the ground. This is problematic for the studies carried out in this chapter because:

- When real F1 cars are travelling over high kerbs it is typical for the inside front wheel to lose contact with the ground.

- When travelling over rough surfaces, the wheel loads will typically be very noisy. Constraining the instantaneous wheel load to zero will result in the mean wheel load having a value greater than zero. Applying a constraint to a noisy value introduces a myriad of constrained local minima into the problem.

As explained in section 2.7.4, the vehicle model equations compute two sets of wheel load values. The first set are the real physical values used to compute the tyre forces. As the wheel unloads, the tyre vertical spring reaches its natural length then when the tyre leaves contact with the ground the force in the spring falls to zero. The real physical tyre load values are at all times greater than zero. We also compute a set of 'virtual' loads. For the computation of the 'virtual loads' the tyre spring is allowed to extend beyond its natural

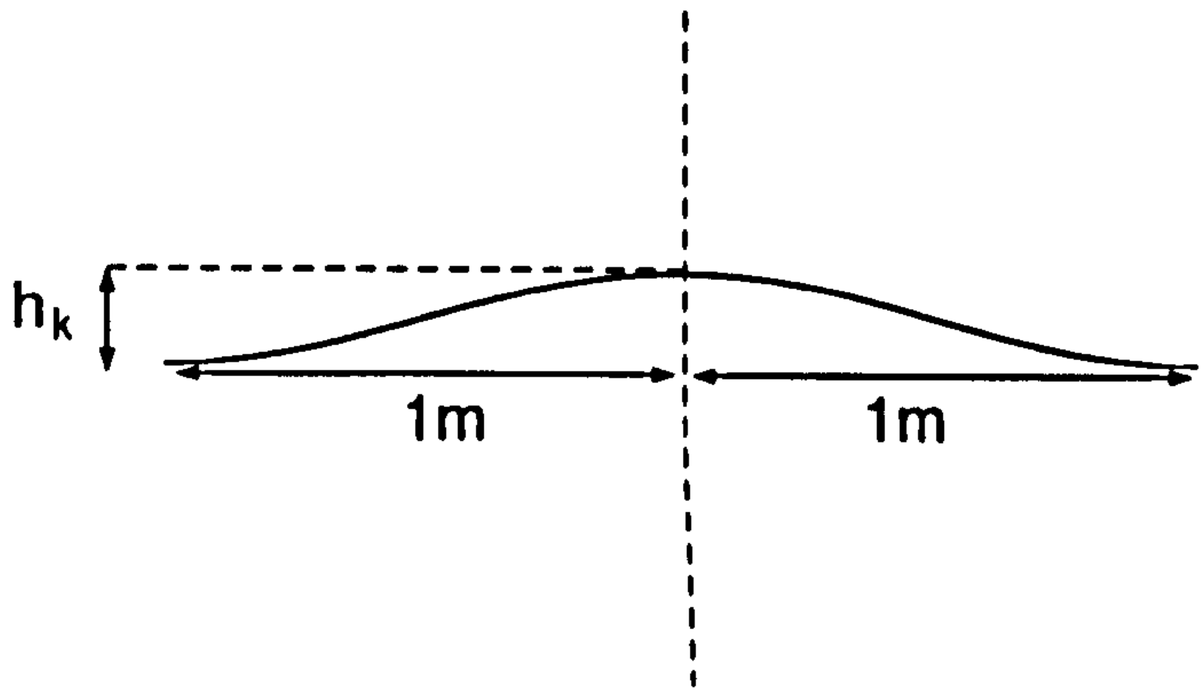


Figure 7.12: Kerb Profile

length. One end of this virtual spring is constrained to remain in contact with the ground at all times hence assuming a negative value when the tyre leaves the ground. In order to address the kerbing problem we modify the wheel load constraint defined in section 2.7.4 as shown in equation 7.11. Instead of limiting the minimum value of the virtual wheel load in between two waylines we limit the average value $F_{zv_{avg}}$. This allows a wheel to leave contact with the ground in optimized solution trajectories. It also goes some way toward reducing the number of local minima introduced by the noisy load constraints. Figure 7.14 shows a typical example of the wheel loads in an optimized solution with a kerb height of 10mm.

$$C_{Fz_{avg}} = F_z^{Lim} - F_{zv_{avg}} \tag{7.11}$$

7.6.1 Roll stiffness

Figure 7.15 shows the manoeuvre time effect of roll stiffness with no kerb and with kerb heights of 5,10 and 15mm. The baseline vehicle setup is shown in table 7.5. The x axis in the figure is the total roll stiffness $K_{roll_F} + K_{roll_R}$. $D_{roll_{spring}}$ is maintained at 0.5. The no kerb plot shows two solution families. The manoeuvre time generally reduces with increased roll stiffness. This is because the car spends less time in roll transients. The 5mm kerb case is similar to the slower of the two no kerb solutions. The 10mm kerb case flattens out at around 50 N/mm. The benefit at higher roll-stiffnesses seen in the no kerb case is clearly lost due to the cost of riding the kerb. In the 10mm case, there is considerable manoeuvre time spread in the results. At this level of roll stiffness, the loads on the unloaded side of the car undergo significant oscillation when riding the kerb. This introduces many local minima into the soultions due to the minimum wheel load constraint. At a kerb height of 15mm, solutions can only be found for values of total roll stiffness less than approximately 70 N/mm (at the heave stiffness values used). Beyond this value, the MTM method cannot find initial feasible solutions for optimization.

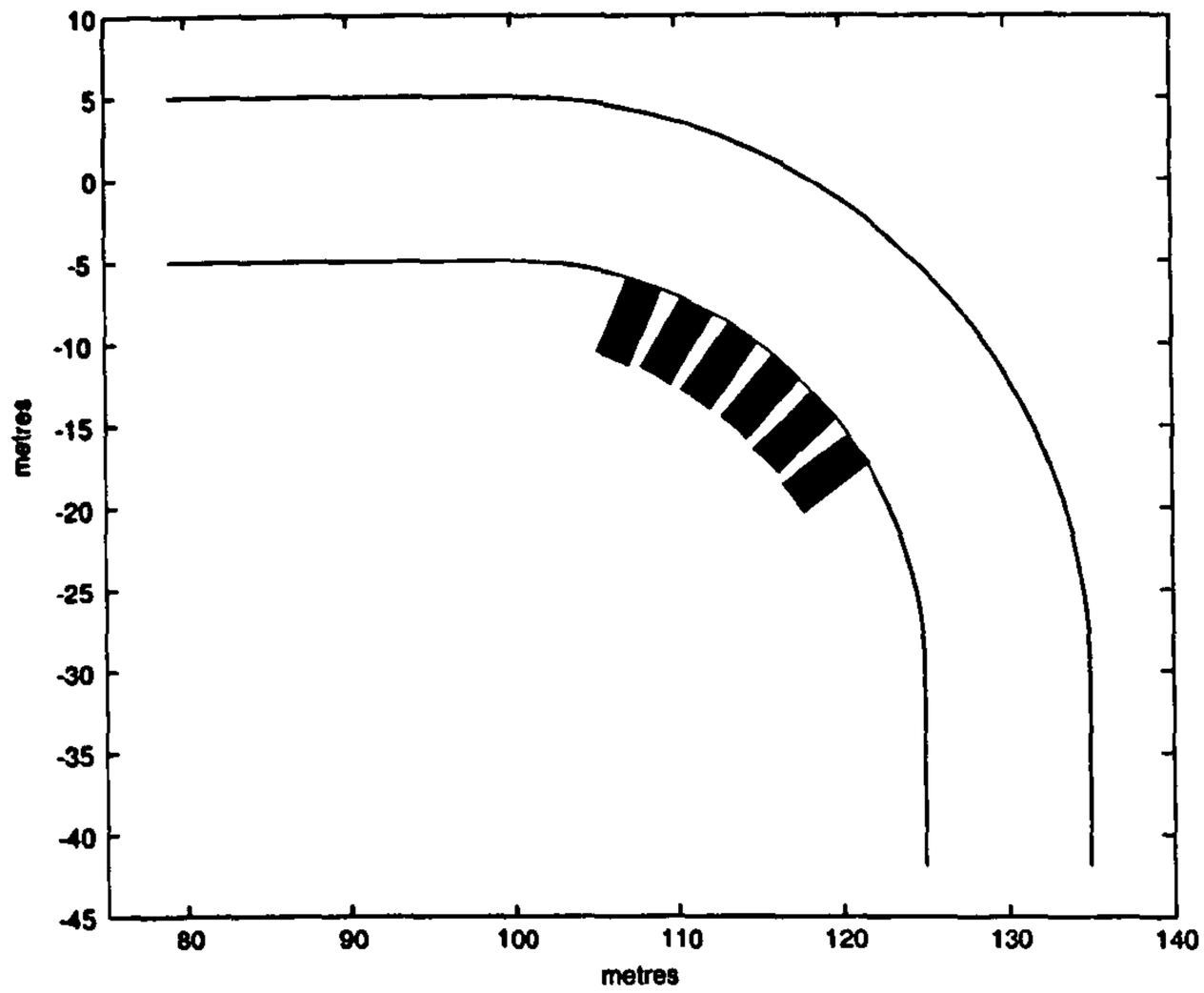


Figure 7.13: Kerb Location

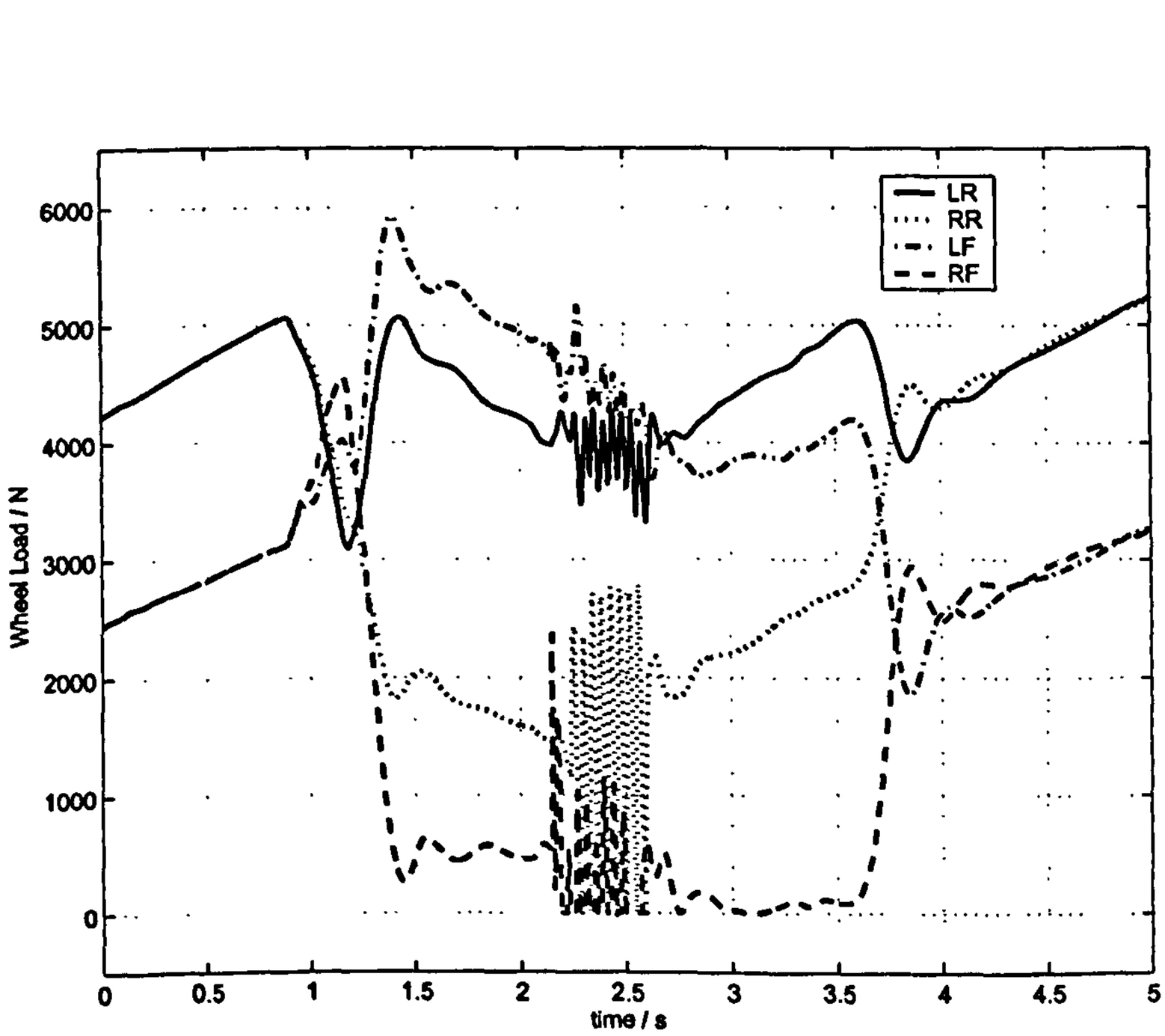


Figure 7.14: Kerbing Wheel loads

Term	Value
K_{heave_F}	60 N/mm
K_{heave_R}	30 N/mm
C_{heave_F}	3 Ns/mm
C_{heave_R}	1.5 Ns/mm
C_{roll_F}	1 Ns/mm
C_{roll_R}	0.5 Ns/mm

Table 7.5: Sprung Vehicle Setup for Kerbing roll stiffness test

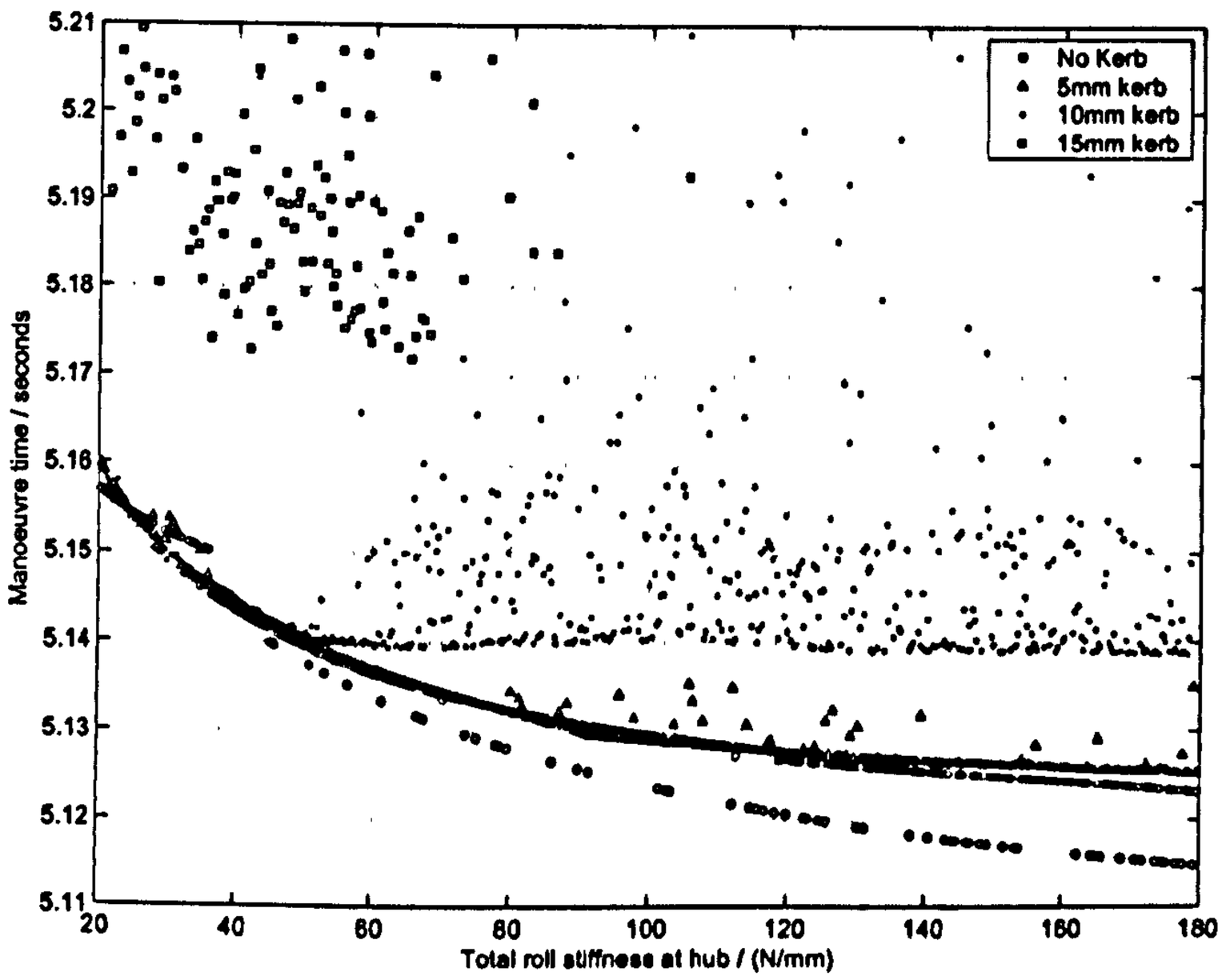


Figure 7.15: Manoeuvre time effect of total roll stiffness with kerb

7.6.2 Heave stiffness

Figure 7.16 shows the manoeuvre time effect of heave stiffness with no kerb and with kerb heights of 5,10 and 15mm. The baseline vehicle setup is shown in table 7.6. The x axis in the figure is the total heave stiffness $K_{heave_F} + K_{heave_R}$. $D_{heave_{spring}}$ is maintained at 0.5. Since only the vehicle heave stiffness is changed with all other parameters maintained, the manoeuvre time increases with heave stiffness largely because of the associated increase in centre of gravity height. The no kerb case can be used as a reference to asses the effect of heave stiffness when kerbing. As expected, when the kerb is introduced the manoeuvre time increases with heave stiffnesses. Considering the 5mm kerb case, there is little time penalty for kerbing up to total heave rates of 70 N/mm. In the 10mm case this reduces to aprroximately 50 N/mm. In the 15mm kerb case, for total heave stiffnesses greater than 120 N/mm there is significant spread in the manoeuvre time results.

Term	Value
C_{heave_F}	3 Ns/mm
C_{heave_R}	1.5 Ns/mm
K_{roll_F}	25 N/mm
K_{roll_R}	12.5 N/mm
C_{roll_F}	1 Ns/mm
C_{roll_R}	0.5 Ns/mm

Table 7.6: Sprung Vehicle Setup for Kerbing Heave stiffness test

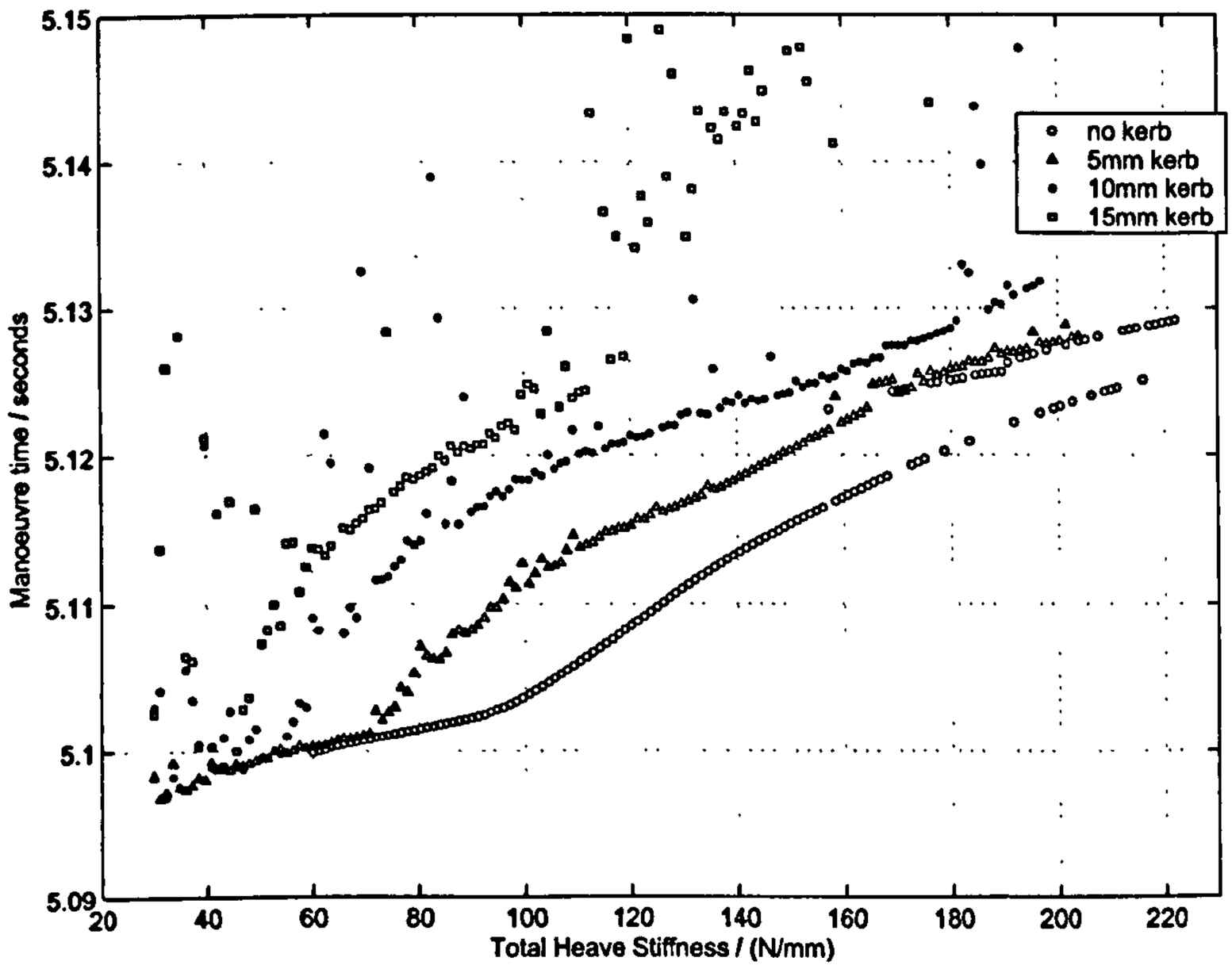


Figure 7.16: Manoeuvre time effect of total heave stiffness with kerb

7.6.3 Roll Damping

Figure 7.17 shows the manoeuvre time effect of roll damping with no kerb and with a kerb height of 10mm. The baseline vehicle setup is shown in table 7.7. The x axis in the figure is the total roll damping $C_{roll_F} + C_{roll_R}$. The ratio of front to rear heave stiffness is maintined as $C_{roll_F} = 2C_{roll_R}$. In the no kerb case, the manoeuvre time decreases with increasing roll damping in a similar manner to the roll stiffness results. There is a fixed offset of approximately 120ms between the no kerb case and the 10mm kerb case. This indicates that there is no penalty for riding the kerb from the increased roll damping.

Term	Value
K_{heave_F}	60 N/mm
K_{heave_R}	30 N/mm
C_{heave_F}	3 Ns/mm
C_{heave_R}	1.5 Ns/mm
K_{roll_F}	30 N/mm
K_{roll_R}	15 N/mm

Table 7.7: Sprung Vehicle Setup for Kerbing roll damping test

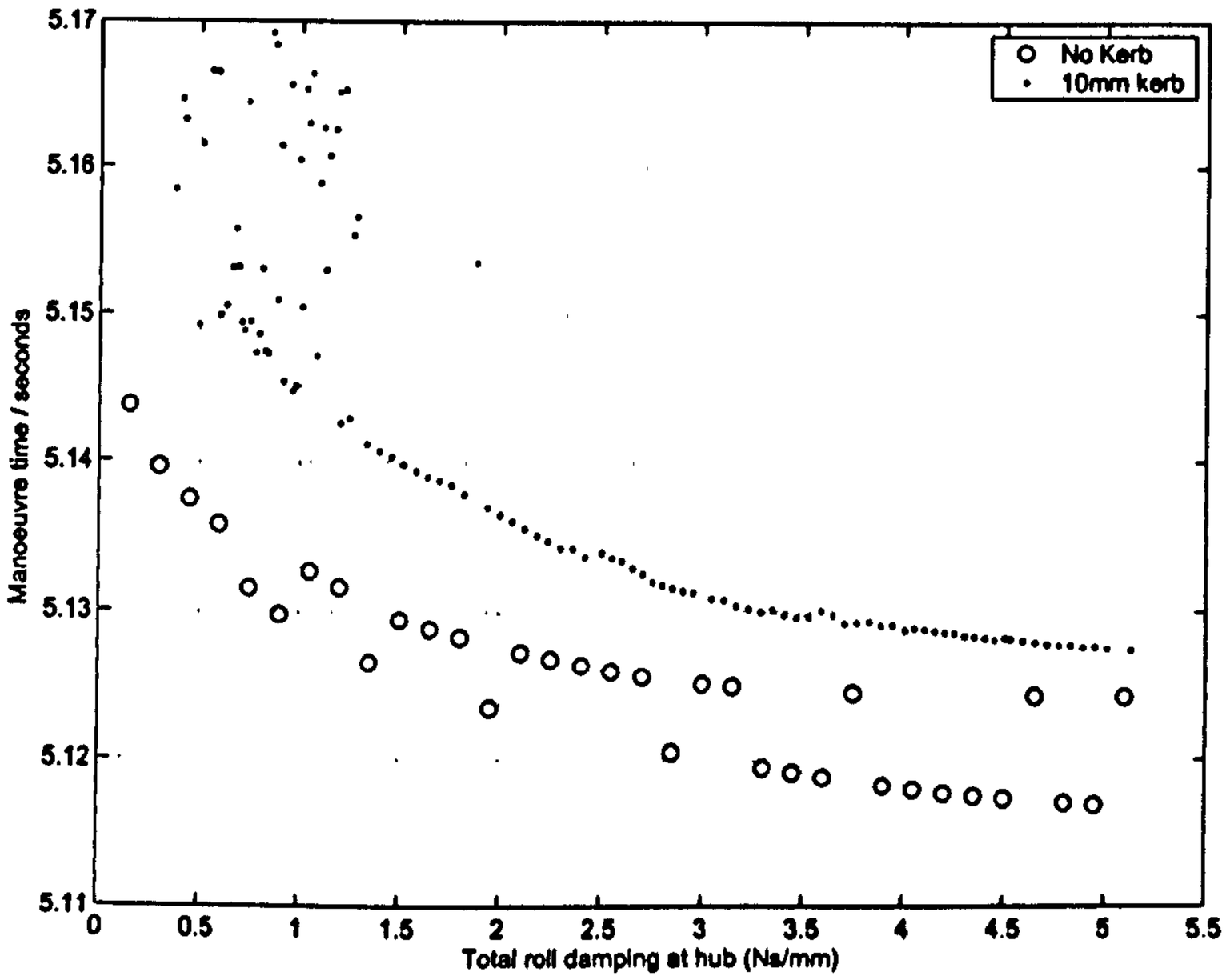


Figure 7.17: Manoeuvre time effect of total Roll damping with kerb

7.6.4 Heave Damping

Figure 7.18 shows the manoeuvre time effect of heave damping with no kerb and with a kerb height of 10mm. The baseline vehicle setup is shown in table 7.8. The x axis in the figure is the total heave damping $C_{heave_F} + C_{heave_R}$. The ratio of front to rear heave stiffness is maintined as $C_{heave_F} = 2C_{heave_R}$. In the no kerb case, there is little change in manoeuvre time with heave damping. The kerb case is considerably more interesting. For stiffnesses below 3 Ns/mm the manoeuvre time is improved with increasing heave damping. Below this value, the vehicle appears underdamped and the wheel load oscillations result in spread in the manoeuvre time solutions. The manoeuvre time achieves a minimum at approximately 3 Ns/mm and beyond that value starts to increase. At approximately 6 Ns/mm the manoeuvre time results start to spread. This is due to the increased effective heave rate when riding the kerb.

Term	Value
K_{heave_F}	60 N/mm
K_{heave_R}	30 N/mm
K_{roll_F}	30 N/mm
K_{roll_R}	15 N/mm
C_{roll_F}	3 Ns/mm
C_{roll_R}	2 Ns/mm

Table 7.8: Sprung Vehicle Setup for Kerbing heave damping test

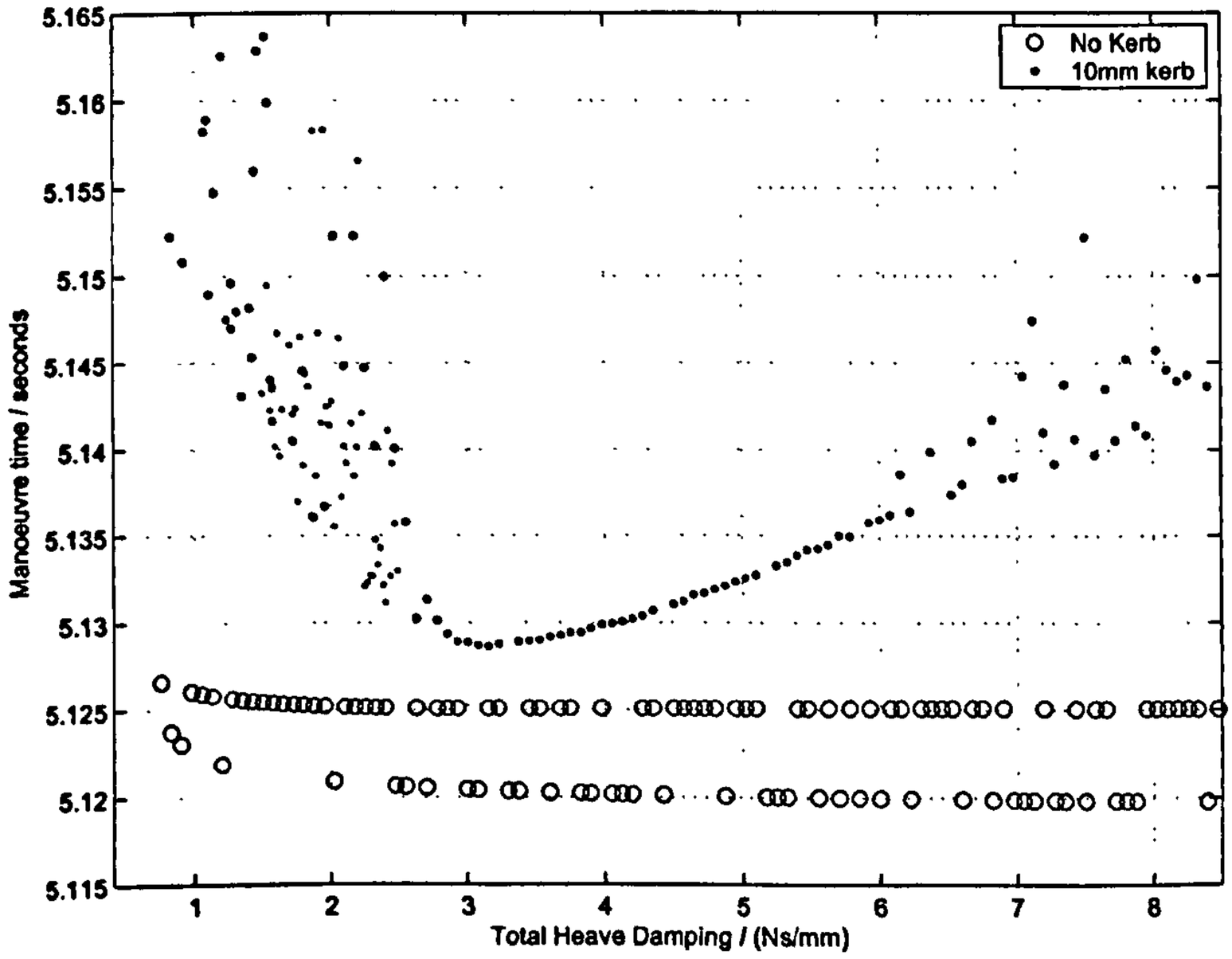


Figure 7.18: Manoeuvre time effect of total Heave damping with kerb

Chapter 8

Performance Analysis using a Thermodynamic Tyre Model

I love the smell of burning rubber

Gilles Villeneuve

8.1 Summary

In this chapter we carry out manoeuvre time optimization using a vehicle equipped with a tyre model that includes dynamic temperature and pressure effects. The tyre model is based on the 'brush' approach [30]. The processes responsible for generating tyre forces are modelled using the physical principles of visco-elastic polymer friction. A simple lumped parameter approach is used for the thermodynamics with bodies for the tyre elements and inflation gas. We firstly examine the effect of tyre temperature on manoeuvre time and vehicle stability for two short manoeuvres at a number of fixed temperatures. We then repeat this experiment with dynamic tyre temperatures across a range of race track surface and ambient temperatures. Finally we examine the lower frequency thermal behaviour using several laps of the Silverstone National Circuit.

8.2 Introduction

The tyre is arguably the most difficult and important aspect of the vehicle model. The part of the tyre that is in contact with the ground is termed the contact patch. It is the friction forces produced at the four contact patches (along with aerodynamic forces) that determine the motion of the vehicle. The physical processes responsible for determining the friction between tyre tread and road are complex. In many categories of racing, the friction developed by the tyre is very sensitive to the temperature of the rubber. Vehicle handling

characteristics and performance can be dominated by tyre temperature effects and this is the motivation for this chapter. Due to the highly competitive and commercial nature of the racing tyre business, tyre related data is difficult to come by and usually must be treated confidentially. The model developed here uses only data that is readily available in the literature. We forego the chance to compare the simulation results obtained with those measured on a real vehicle in order to allow unrestricted access to this work. The purpose of this chapter is therefore largely to seek to demonstrate that the developed MTM method is capable of operation with a tyre model including thermal effects. Whilst we do simulate the effect of certain parameters such as tyre and track temperature on manoeuvre time, no claim is made that the results obtained are highly correlated to any specific vehicle and tyre. The author does however believe that the developed tyre model is useful and that the results provide an insight into the impact of tyre temperature effects on vehicle handling and performance.

In simple terms, the tyre can be considered to consist of a 'carcass' and a 'tread'. The carcass gives the tyre its basic shape and mechanical strength. It is typically of a complex woven construction [20][17]. It will typically contain a 'belt' that is responsible for supporting the tread. The sidewalls of the carcass are responsible for supporting the belt and form part of the suspension of the sprung mass for the vehicle. In racing terms, the carcass is usually termed the tyre 'construction' and the tread rubber the 'compound'. As the tyre rotates, the tread compound enters and leaves the contact patch. During this process, it stretches and relaxes. While in the contact patch the tread may be sliding over the non-smooth road surface, which again causes the tread to be deflected. The carcass also undergoes a cycle of stretching and relaxation during rotation as it deflects in the process of supporting the vertical load and supplying lateral and longitudinal forces. Rubber is a visco-elastic material and in this stretching and relaxing process, the work done is converted to heat. This heat generation process, along with heat flows between tread, road, carcass, inflation gas, tyre rim and surrounding air determine the compound temperature. What follows is the development of a tyre model based on the brush concept. The model includes a much simplified case of these thermal effects.

The brush model [30] considers the tyre tread as a row of elastic bristles. We will consider that one end of each bristle is held in a fixed position by the belt. As the tyre rotates, the bristles enter the contact patch and the end of the bristle in contact with road is deflected due to the relative motion of the belt and road. The tread deflects until the friction force between bristle and road surface is unable to support any further deflection and the tread begins to slide relative to the road surface. The contact patch is considered as two distinct regions: sliding and non-sliding.

The process by which the friction forces are developed is complex. The road surface contains asperities both visible to the naked eye and at the microscopic scale. The tyre tread rubber is elastic and deforms in contact with the road surface. Hence the contact between the tread and ground consists of many microscopic contacts. The coefficient of friction between rubber and a considerably less compliant medium (e.g. road surface) is generally considered to be a function of:

1. Temperature of the rubber

2. Pressure in the contact

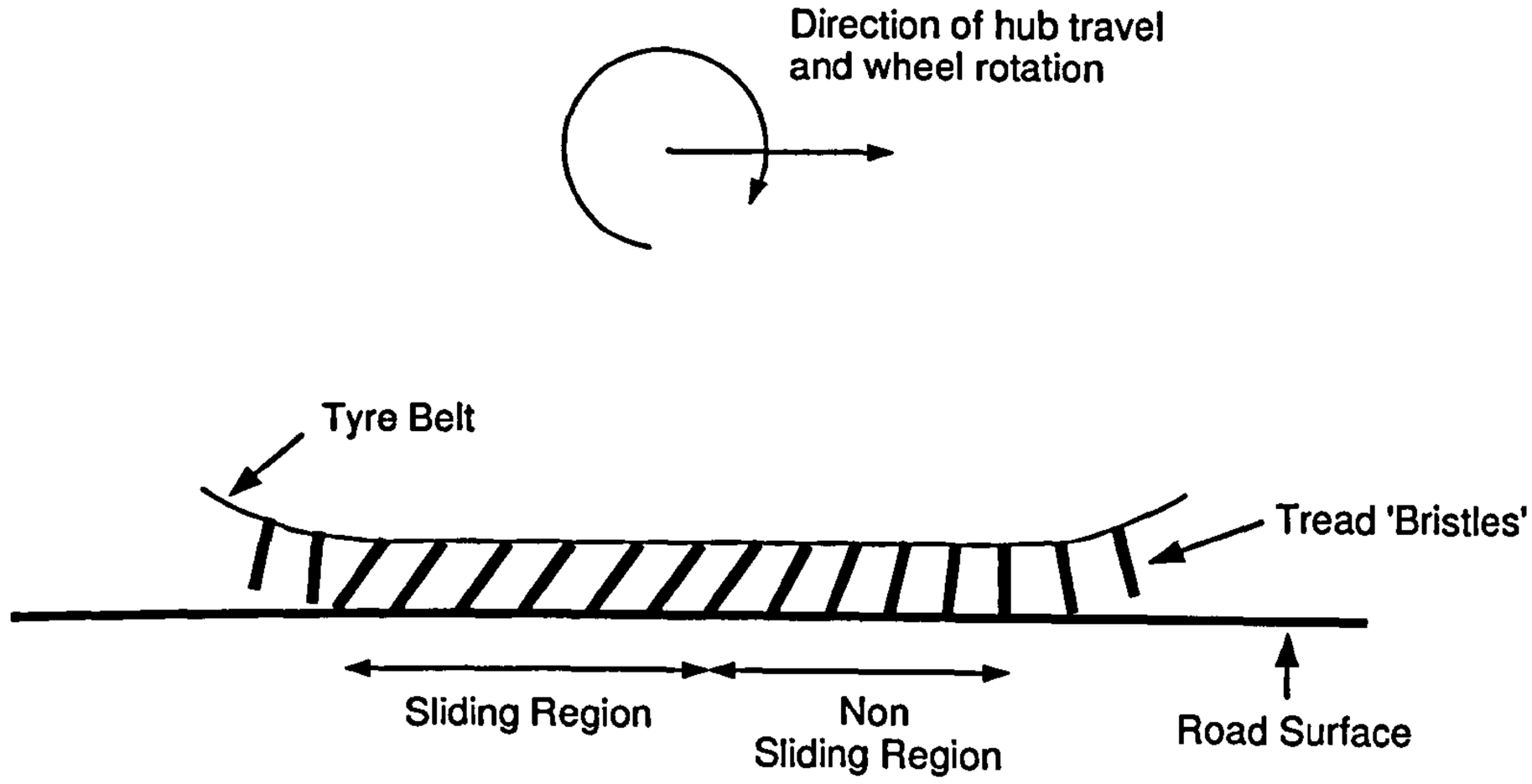


Figure 8.1: Brush Model

3. Sliding velocity
4. Lengths associated with the surface roughness

The pressure in the contact patch is a function of the vertical load on the tyre and the contact patch geometry. For a given tyre construction, we assume the contact patch area is influenced by:

1. Vertical Load
2. Inflation pressure

In practice the contact patch shape will also be influenced by distortions in the carcass when generating forces. At a simple level, the average temperature of the tread can be considered to be a function of:

1. The self heating of the rubber (viscous damping)
2. Heat transfer between road and tread (conduction)
3. Heat transfer between air and tread (convection)
4. Heat radiated from tyre
5. Heat transfer between tread and carcass

The brush approach is centred around modelling tread deflection relative to the belt. For our model we assume that the belt undergoes no longitudinal or lateral deflection. Tyre

constructions typically consist of a woven composite of a number of materials such as steel and nylon and as such are very difficult to model. A number of finite element models have been developed in the literature [31] but such models are far too computationally intensive to be used in a lap time simulation program. In practice, the aligning moment present on a tyre when generating lateral force causes the tyre carcass to deflect and this has a significant effect on the μ/slip behaviour. We neglect this in this work though it is a relatively easy task to add this type of behaviour to the model.

8.3 Thermal Brush Model

The basic brush model described below follows that given by Pacejka [30]. The diagram below shows a schematic for the contact patch where the x axis represents the longitudinal position of a point in the contact patch. The origin for these axes is the centre of the contact patch. The contact patch length is given by $2a$. Pacejka introduces two practical slip quantities as given in equations 8.1 and 8.2.

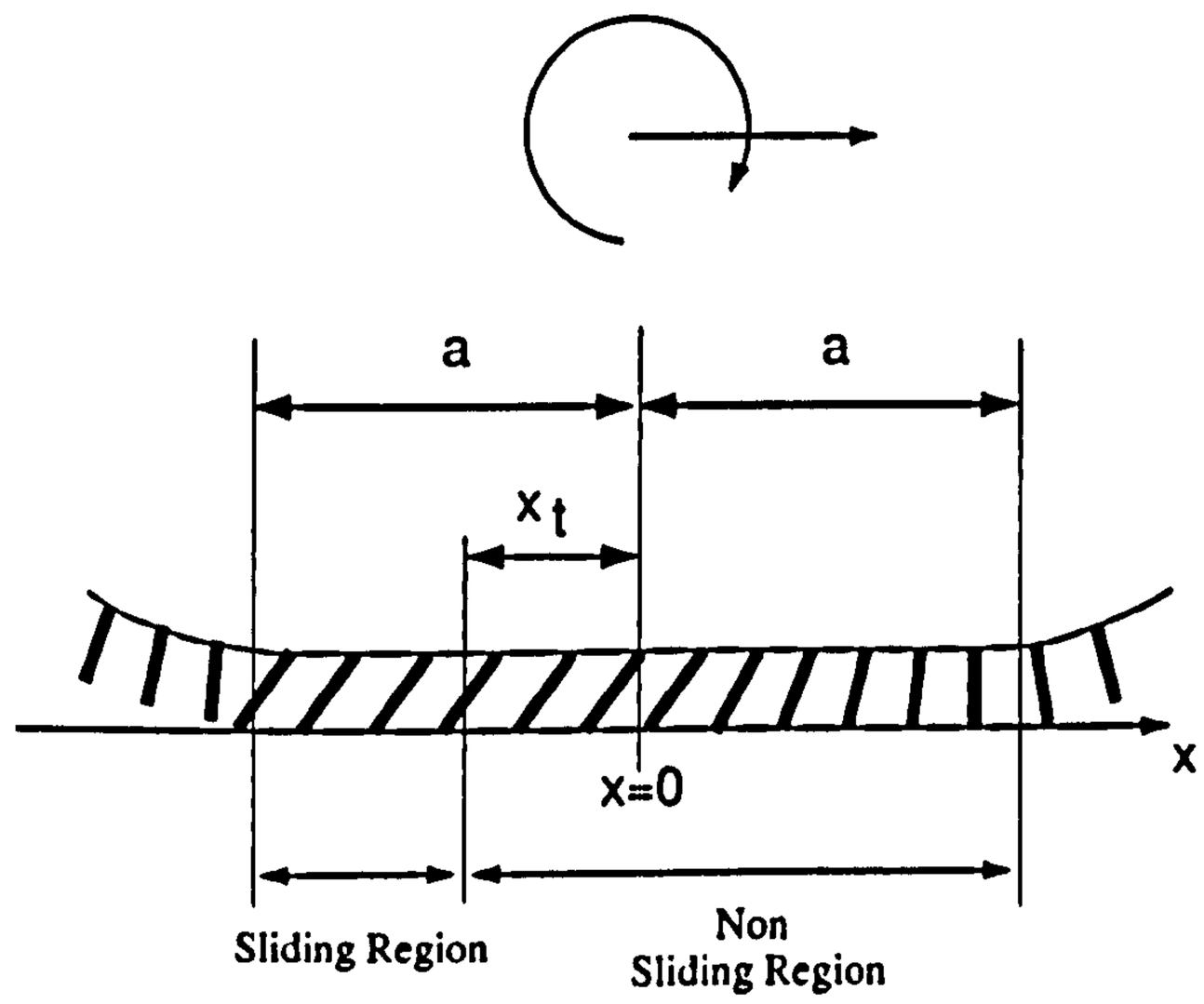


Figure 8.2: Brush Model

$$\sigma_x = \frac{\kappa}{1+\kappa} \tag{8.1}$$

$$\sigma_y = \frac{\tan(\alpha)}{1+\kappa} \tag{8.2}$$

$$\bar{\sigma} = \begin{bmatrix} \sigma_x \\ \sigma_y \end{bmatrix} \tag{8.3}$$

Where κ and α are the slip ratio and slip angle in traditional form [30]. When the tyre is in slip, a point on the belt has a non-zero velocity relative to the ground speed of the wheel hub. This relative velocity leads to a linearly increasing deflection from zero deflection at the leading edge of the contact patch. The vector tread element deflection in the non-sliding region is expressed using the practical slip quantities as shown in equation 8.4.

$$\bar{e} = (a - x)\bar{\sigma} \tag{8.4}$$

The vector contact force (per unit contact length) is given in equation 8.5.

$$\bar{q} = k_c(a - x)\bar{\sigma} \tag{8.5}$$

Where k_c is the stiffness of the tread bristles per unit length (along x axis) of tread. This is a fundamental property of the compound and is related to the shear modulus of elasticity of the material. We assume here that this property is isotropic (equal for deflections in all directions in the xy plane). We define k_c in equation 8.6 in terms of tread compound sheer modulus G_{tread}, tread height h_{tread} and contact patch width w_{cp}.

$$k_c = \frac{w_{cp}G_{tread}}{h_{tread}} \tag{8.6}$$

If $q_z(x)$ is the vertical load on the contact patch at point x and μ_0 is the static coefficient of friction, then sliding begins at the point where the static friction force can support no further deflection as described by equation 8.7.

$$\|\bar{q}\| = \sqrt{q_x^2 + q_y^2} > \mu_0 q_z(x) \tag{8.7}$$

$$\|k_c(a - x_t)\bar{\sigma}\| = \mu_0 q_z(x) \tag{8.8}$$

The solution to Eqn 8.8 provides the point x_t along the x axis where sliding begins. Assuming the expression for $q_z(x)$ is sufficiently complex, this equation may be solved iteratively using the Newton-Raphson procedure. Given x_t and a it is then a simple procedure to compute the tyre forces as the sum of those produced in the sliding and non-sliding regions. The lateral force is given by:

$$F_y = \int_{-a}^{x_t} \frac{\sigma_y}{\|\bar{\sigma}\|}\mu_d q_z(x).dx + \int_{x_t}^{a} k_c(a - x)\sigma_y.dx \tag{8.9}$$

The longitudinal force is given by:

$$F_x = \int_{-a}^{x_t} \frac{\sigma_x}{\|\bar{\sigma}\|}\mu_d q_z(x).dx + \int_{x_t}^{a} k_c(a - x)\sigma_x.dx \tag{8.10}$$

Where μ_d is the dynamic or sliding coefficient of friction.

8.3.1 Contact Patch Dimensions

The length and width of the contact patch when a tyre is running over a flat surface in zero slip conditions (and with no camber) are typically functions of inflation pressure and vertical

load. A tyre supports its vertical load through the gas pressure acting above the contact patch and through the stiffness of the sidewall. Contact patch length increases with vertical load and generally decreases with inflation pressure. We assume that the front and rear tyres have differing constructions hence differing expressions for contact patch width and length. The expressions used to describe the contact patch lengths are given in equations 8.12 and 8.11.

$$2a_R = 0.132\, p_{gbarg}^{-0.7} \left(\frac{F_z}{3000}\right)^{0.7} \tag{8.11}$$

$$2a_F = 0.114\, p_{gbarg}^{-0.7} \left(\frac{F_z}{3000}\right)^{0.8} \tag{8.12}$$

We assume that the belt is sufficiently rigid such that the contact patch width does not change significantly with vertical load or inflation pressure. The contact patch widths are given in equations 8.14 and 8.13.

$$w_{cp_R} = 0.225 \tag{8.13}$$

$$w_{cp_F} = 0.205 \tag{8.14}$$

We make the simplifying assumption that the contact patch is rectangular. In practice it will be of a more complex shape and will be affected by distortions in the carcass brought about by lateral and longitudinal forces, wheel camber and the non-smooth nature of the road surface. Equations 8.15 to 8.17 define the contact patch area and pressure.

$$A_{cp_R} = 2\, a_R\, w_{cp_R} \tag{8.15}$$

$$A_{cp_F} = 2\, a_F\, w_{cp_F} \tag{8.16}$$

$$P_{cp} = \frac{F_z}{A_{cp}} \tag{8.17}$$

8.3.2 Contact Patch Vertical force Distribution

In [30] Pacejka assumes that the contact patch pressure distribution is parabolic. This is a reasonable assumption for road car tyres and allows easy solution of Eqn (6). Racing tyres are usually designed to have large contact patches. This is achieved through properties of construction and relatively low running tyre pressures. This results in a pressure distribution that is quite constant along the contact patch. The equation for $q_z(x)$ given by Pacejka is shown in equation 8.18. Pacejka assumes a value of $n = 2$ to produce the parabolic distibution. We modify this to a value of 4 to flatten the pressure distribution as shown in figure 8.3.

$$q_z(x) = \frac{(n+1)F_z(a^n - x^n)}{2na^{n+1}} \tag{8.18}$$

This simple contact patch pressure distribution is probably only representative of the tyre under static conditions. When the tyre is rotating the pressure distribution is likely to

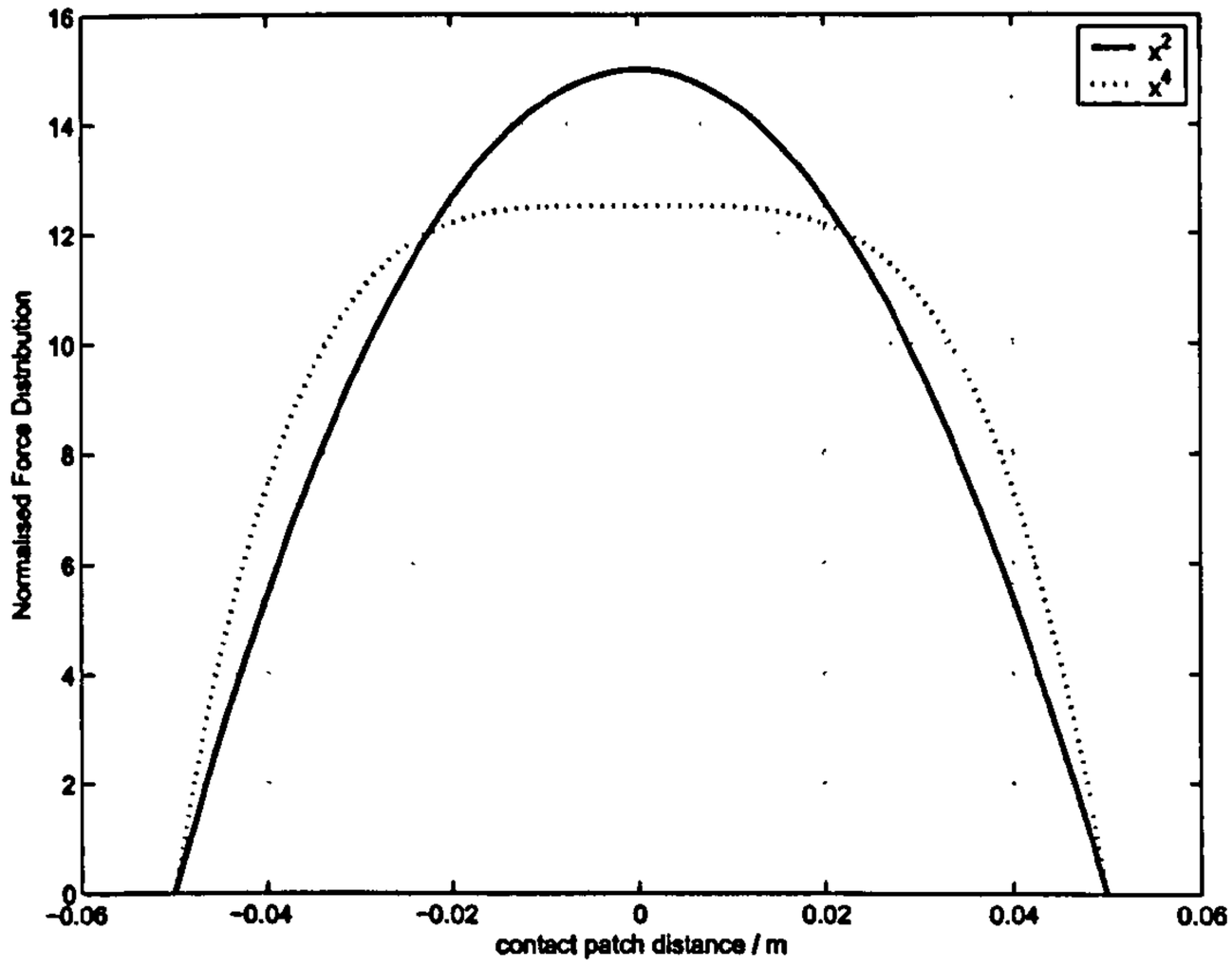


Figure 8.3: Normalised Force Distribution

be more complex. Fujikawa et. al. [48] show through measurements that the pressure in the leading edge of the contact patch is higher than the trailing. Haney [43] discusses the effect of 'tread momentum' on contact patch pressure. The tread and carcass entering the contact patch must be forced to change its direction away from its steady state rotational path. The force to provide this acceleration is seen in the contact patch.

8.3.3 Rubber Friction

Rubber is a visco-elastic material. As the tyre contact patch slides over the road surface the friction produced is closely related to this visco-elastic property [29]. The visco-elastic properties of the polymers used in tyre rubber are dependent on both temperature and excitation frequency [18],[29],[23],[42],[40]. The excitation frequencies associated with the sliding process are related to the surface roughness of the substrate on which the sliding takes place. Consider first rubber sliding over a smooth substrate. Adhesive bonds are formed between molecules in the polymer and molecules in the substrate, the bonded rubber is then stressed by the sliding until the bonds are broken. This process takes place repeatedly and the stress and relaxation of the rubber results in energy loss due to viscous damping.

Considering each molecule in the rubber as an ideal spring in parallel with an ideal viscous damper. After a bond is made the spring will start to extend until the force required exceeds the strength of the bond. When the bond is torn apart the spring will 'recoil' before extending again to reform a bond with the substrate. The excitation frequency is hence dependent on the sliding velocity and the distance between successive bonds. Secondly, when sliding over a rough substrate the rubber will 'key' into the asperities and the rubber must be displaced from the gaps in between the asperities during sliding. The velocity imposed on the rubber by this displacement also results in viscous damping losses. The excitation frequencies associated with this process are dependent on the sliding velocity and

on the profile of the surface. The typical road surface will have asperities on many different scales from 'macro' roughness due to the size of particles through to 'micro' roughness when considering the surface roughness of the particles themselves. Persson [42],[40],[41] develops a method for analysing friction where the surface roughness has a self affine fractal structure.

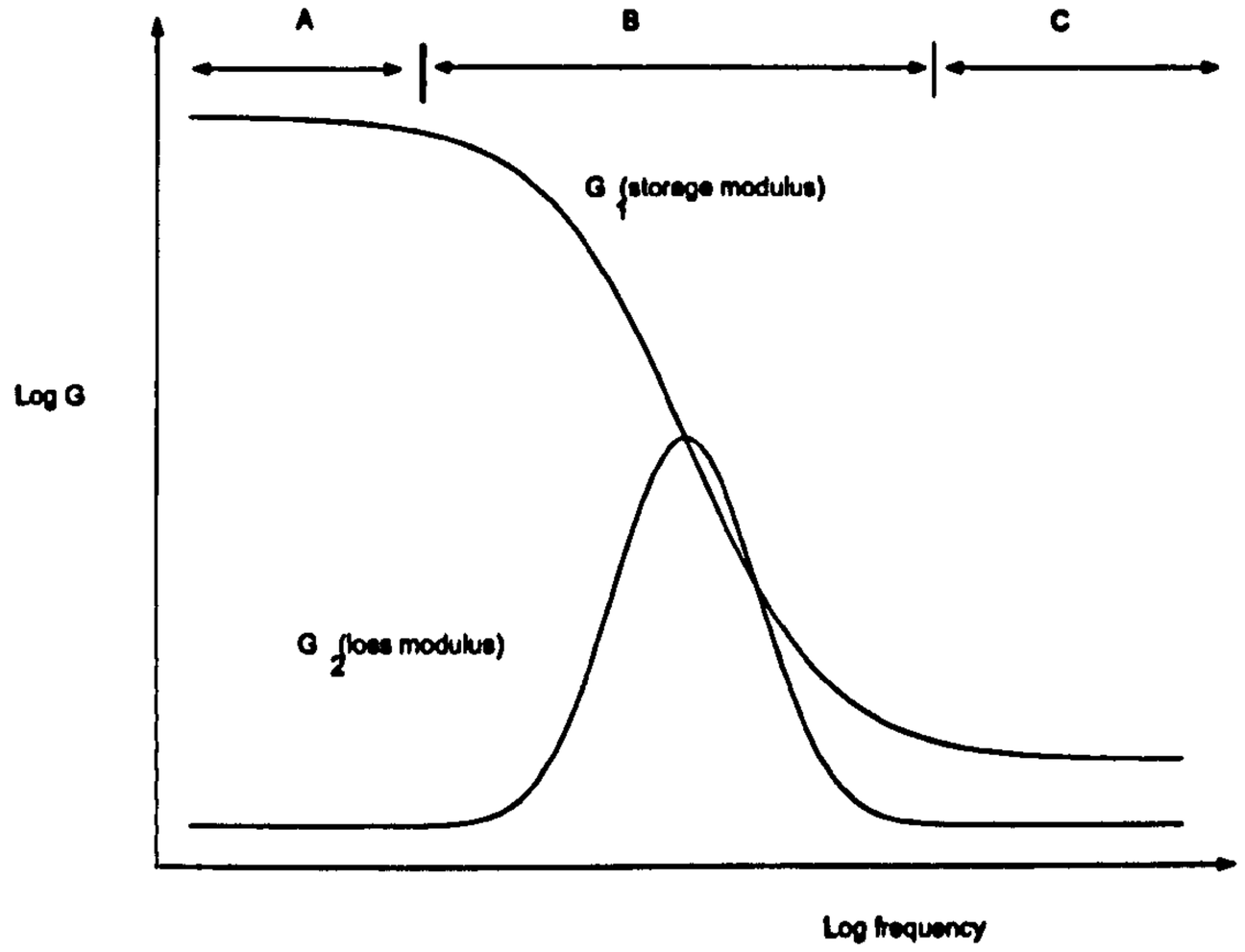


Figure 8.4: Polymer Loss G_1 and Storage Modulus G_2

Figure 8.4 shows the storage and loss moduli G_1 and G_2 of the type of polymer typically found in tread rubber. Region A is known as the glassy region. In this region, the material is stiff and offers little hysteresis. Region C is known as the rubbery region. Region B is a transition region and it is here that the loss modulus of the material achieves its maximum value. Grosch [29] shows that Rubber friction is closely related to the loss modulus and that friction displays the same behaviour with respect to frequency. Williams, Landel and Ferry have shown [18] that there is a 'time-temperature' equivalence for polymers of this type and the x axis of figure 8.4 can be replaced by temperature and a suitable scale factor. In practice, the loss modulus or dynamic coefficient of friction of rubber can be defined for all frequencies and temperatures using a single 'master curve'.

The master curve describes the coefficient of friction against log frequency (or sliding speed for a given surface) at a given reference temperature. The curve can be referred to other temperatures by a shift along the (log) frequency axis. An example of this is shown in figure 8.5. We show the equivalent curves on a linear axis in figure 8.6.

We assume that the master curve for the tread compound has a Gaussian shape and adopt the expression given in equation 8.19 for the dynamic coefficient of friction where v_s is the sliding velocity and T_{tread} is the temperature of the tread. The factors μ_{base}, μ_{peak}, K_{shape}, K_{shift} and T_{REF} are used to calibrate the model.

$$\mu_{mc} = \mu_{base} + (\mu_{peak} - \mu_{base}) e^{-(K_{shape}(log10(v_s) - K_{shift}(T_{tread} - T_{REF})))^2} \tag{8.19}$$

We assume also that the coefficient of friction decreases linearly with contact patch

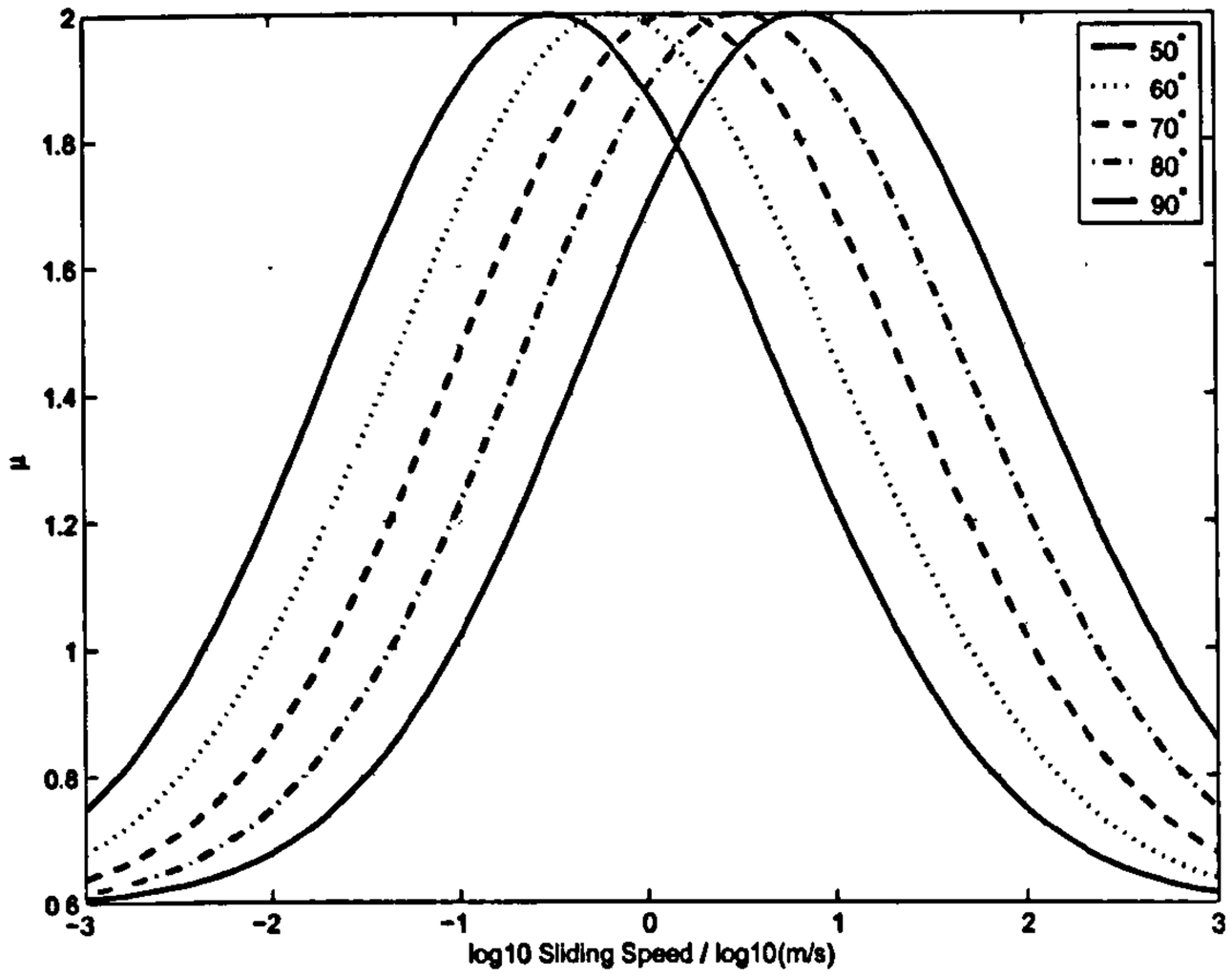


Figure 8.5: Example Dynamic/Sliding μ Curves

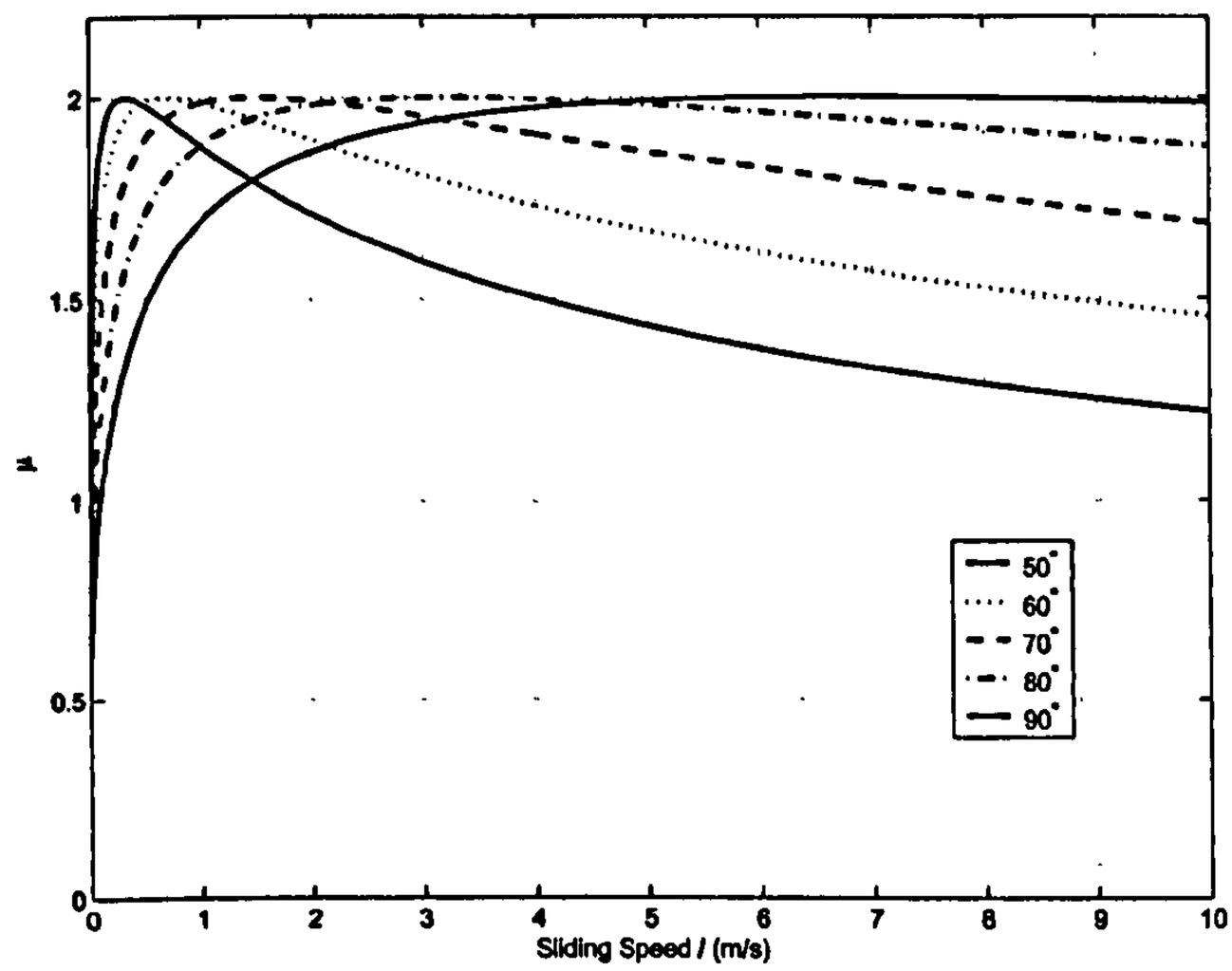


Figure 8.6: Sliding μ

pressure by applying the scaling factor given in equation 8.20 to produce the final expression for the dynamic coefficient of friction given in equation 8.21.

$$K_{cpp} = 1.0 - K_{pcpp}\frac{P_{cp}}{K_{refcpp}} \tag{8.20}$$

$$\mu_d = K_{cpp}\mu_{mc} \tag{8.21}$$

For the static coefficient of friction we assume a simpler model that is independent of temperature as given in equation 8.22. In practice it may be temperature dependent.

$$\mu_0 = K_{cpp}\mu_{REF} \tag{8.22}$$

Tyre tread rubber is typically a compound of many polymers and other substances such as fillers. The master curve for such a compound sliding over a complex surface is likely to be more complex than that used here and may contain more than one peak. The process by which rubber compounds are manufactured is complex. The manufacturing process consists of several stages in which substances are added and the compound is mixed and heated. This process forms links between the polymer molecules and distributes filler material particles through the compound. In the case of the high performance racing tyre the thermal cycling encountered when the tyre is operating on the racing circuit can be viewed as an extension of this process. Tyres may need one 'out' lap to complete the curing process before giving maximum performance for a single lap and then starting to deteriorate. In addition, racing tyres can contain quite volatile substances which one would expect to boil off the tyre in operation. The shelf life of soft racing tyres is short. Stored tyres will harden through interaction with pollution in the environment and sunlight. It seems reasonable therefore to assume that the master curve for a racing tyre will change during its operation on the track. This will not be considered further here.

8.3.4 Tread Rubber Shear Modulus

The shear modulus of rubber is related to its storage modulus G_1 as shown in figure 8.4. In the visco-elastic region the $log(G_1)$ reduces approximately linearly with log frequency. We adopt the expression for compound shear modulus given in equation 8.26. The parameters G_{TA} and G_{TB} represent reference shear modulus values at temperature T_{GA} and T_{GB} with G_{base} representing the limit shear modulus value at high temperature. An example shear modulus against temperature curve is given in figure 8.7.

$$K_A = G_{TA} - G_{lim} \tag{8.23}$$

$$K_B = G_{TB} - G_{lim} \tag{8.24}$$

$$K_G = \frac{log(K_A) - log(K_B)}{T_{GB} - T_{GA}} \tag{8.25}$$

$$G_{tread} = \frac{K_A}{e^{-K_G T_{GA}}} e^{-K_G T_{tread}} + G_{base} \tag{8.26}$$

In practice, the shear and loss moduli are related to the same basic physical properties of a material and are hence not independent. In our model we rely on suitable calibration of the shear and loss moduli expressions to reflect this.

8.3.5 Tyre Heat Generation

We will simplify the consideration of heat generation by considering only two sources:

1. Viscous losses due to deflection of the tyre carcass when producing vertical, lateral and longitudinal forces.
2. Viscous damping in the sliding region.

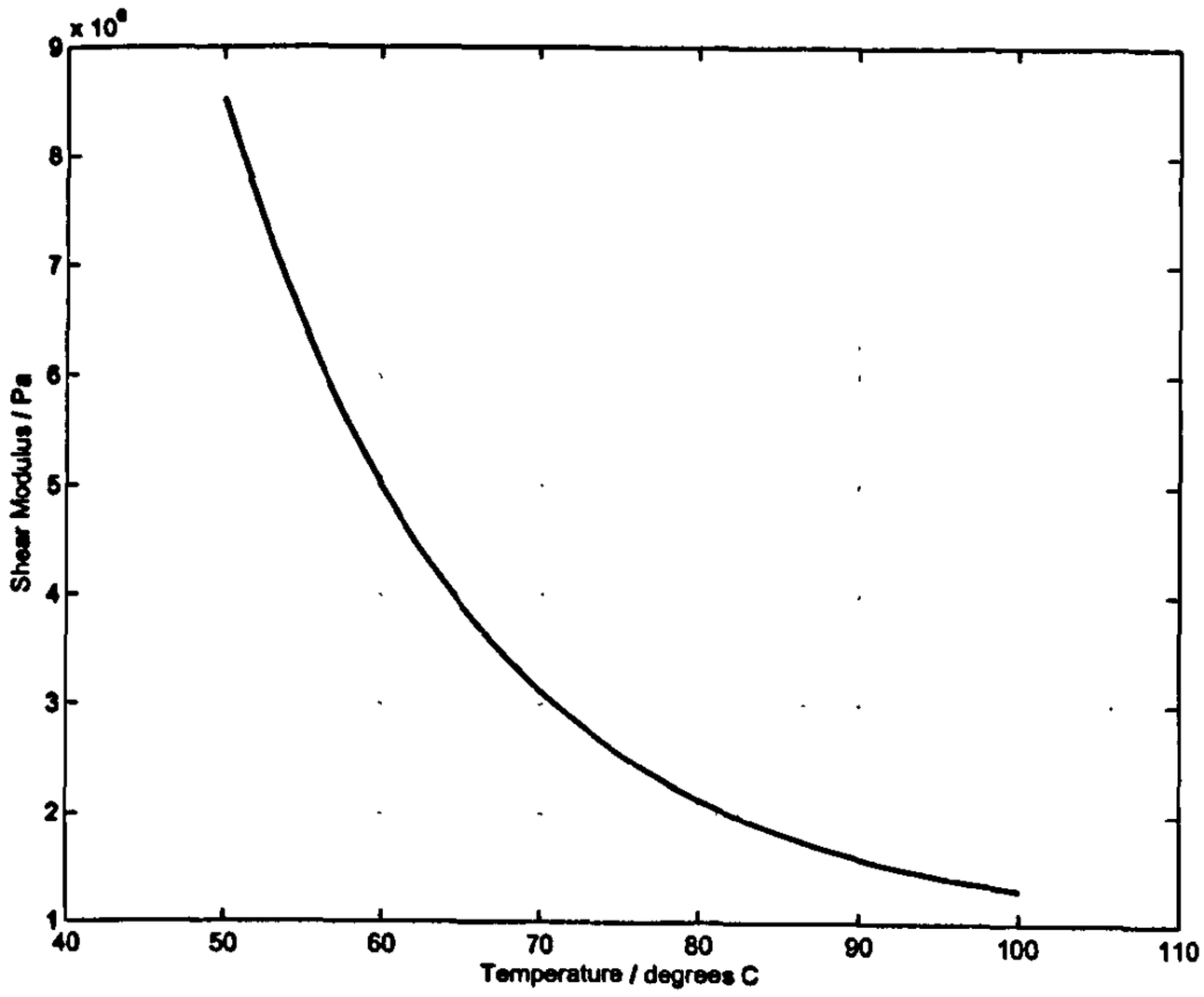


Figure 8.7: Example Shear Modulus temperature effect

Other heat sources such as viscous damping of the general tread deflection are neglected. This is certainly an oversimplification of the heat generation process and further development in this area would be useful.

8.3.5.1 Heat Generation from tyre bulk deflection

The tyre supports the wheel load through the inflation gas acting as a spring and through the stiffness of the tyre sidewalls and belt. As the tyre rolls, the sidewall and belt are deflected as they enter and leave the contact area. Heat is hence produced due to losses in the viscoelastic materials making up the tyre. This corresponding power loss is known as the 'rolling resistance' of the tyre. Simply rolling a loaded tyre along the ground in the absence of slip requires a longitudinal force to overcome these losses. The modelling of rolling resistance has been studied extensively in the literature [59] for passenger cars and trucks as it is a significant factor in both fuel economy and tyre failure. As previously stated, tyre constructions are complex and a truly representative model of rolling power loss is beyond the scope of this work. In addition to the vertical deflection, the tyre undergoes deflection when generating longitudinal and lateral forces. We model the power generated from these sources as simple 'efficiency' terms.

$$Q_{F_z} = v_x \, E_z \, |F_z| \tag{8.27}$$

$$Q_{F_y} = v_x \, E_y \, |F_y| \tag{8.28}$$

$$Q_{F_x} = v_x \, E_x \, |F_x| \tag{8.29}$$

In practice, rolling losses in tyres reduce with temperature [31]. This has a stabilising effect, since as the tyre heats its ability to further heat itself is reduced. We do not account for this fact in this model. We also assume that any work done on the inflation gas as the tyre compresses and relaxes in rotation is negligible.

8.3.5.2 Heat Generation from tread sliding

As the tread rubber slides over the road we assume that all of the work done is converted to heat. The heat due to sliding is show in equation 8.30 where v_s is the magnitude of the velocity between contact patch and road.

$$Q_{sliding} = \mu_d F_z v_s \tag{8.30}$$

In practice, only part of the tyre vertical load is carried by the sliding part of the contact patch. We use the full load to attempt to compensate for lack of any model of the heat generated in the non-sliding region.

8.3.6 Heat Flow

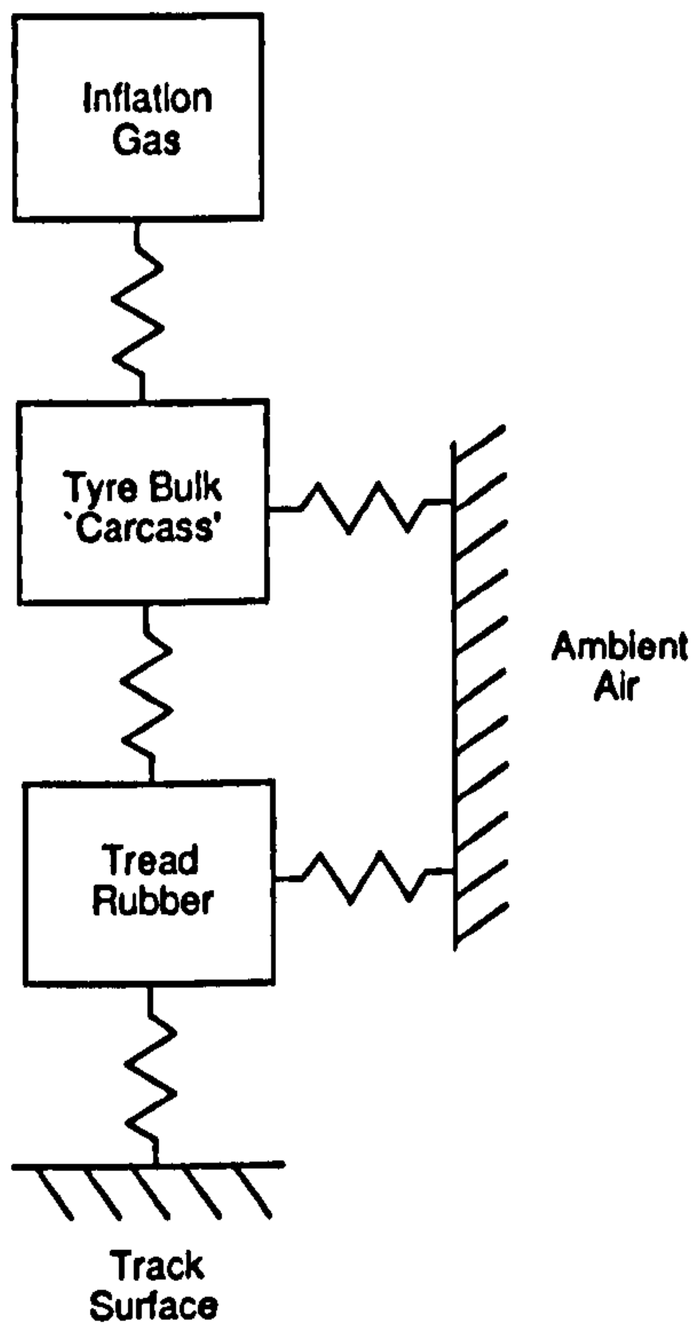


Figure 8.8: Thermal Schematic

For simplicity we represent the tyres thermal behaviour using a lumped parameter approach consisting of the folllowing three bodies:

- The tread with temperaure T_{tread}
- The bulk or carcass with temperature $T_{carcass}$
- The inflation gas with temperature T_{gas}

We consider the track surface and surrounding air as representing boundary conditions with fixed temperarures T_{road} and T_{amb}. We make the simplifying assumption that the 'tread' has a single temperature value. In practice the heating and cooling in the contact patch and the further heat transfer outside the contact patch result in a complex temperature profile for the tread. Fujikawa et. al [48] show through modelling and measurement in a real tyre contact patch that the temperature increases approximately linearly through the sliding region. In [41] Persson shows that local heating effects can have a significant effect on friction.

8.3.6.1 Inflation Gas Pressure

We assume that the volume occupied by the inflation gas is constant. The tyre pressure is typically set at the start of a simulation to a given value at a given gas temperature. The mass of gas in the tyre is computed at this point and used for all subsequent pressure calculations. The gas pressure is computed from the gas temperature, mass and volume using the ideal gas law. Tyres used in formula one racing are typically filled with dry air or Nitrogen, we will assume Nitrogen. The molar mass of Nitrogen is 28.014 g mol-1. We assume that the specific heat capacity of Nitrogen is independent of temperature and pressure with a nominal value of 1.042 kJ Kg-1 K-1. In the presented results, units of Bar gauge are used for inflation pressure.

8.3.6.2 Heat Transfer with surrounding air

Under practical operating circumstances the tyre will be cooling into the surrounding air by convection and radiation. The air flow around the tyre of an open wheeled race car is complex and likely to be turbulent. Browne [5] carries out some wind tunnel testing taking into account both speed, turbulence and surface wetness. We model the heatflow between tread and air as given in equation 8.31.

$$Q_{tread->amb} = H_{tread-amb}(T_{tread} - T_{amb}) \tag{8.31}$$

The heat transfer coefficient $H_{tread-amb}(v_x)$ is assumed to be a function of the longitudinal car speed v_x. For simplicity we assume that it is a simple linear function of v_x as given in equation 8.32.

$$H_{tread->amb}(v_x) = 2v_x + 10 \qquad \frac{W}{m^2K} \tag{8.32}$$

We assume that the carcass can also transfer heat with the ambient air as described in equation 8.33. In this case we assume a fixed value for heat transfer coefficient $H_{carcass-amb}$.

$$Q_{carcass->amb} = H_{carcass-amb}(T_{carcass} - T_{amb}) \tag{8.33}$$

The tyre also loses heat due to radiation and it seems reasonable to assume that the tyre is close to a being black body source. Brown [6] carries out some practical testing of tyre emmissivity ϵ and concludes that a value of 0.94 is representative. In practice, the environment (track and air) also radiate heat which may be absorbed by the tyre. Also, the carcass and inflation gas must also transfer heat by radiation. In practice, these effects result in very small heatflows when compared to other factors. For simplicity we neglect heat transfer through radiation in this model.

8.3.6.3 Heat Transfer with the ground

Heat transfer between tread and road is considered to be a function of the contact patch area A_{cp} as shown in equation 8.34 where $H_{tread-road}$ is the heat transfer coefficient per unit area.

$$Q_{tread->road} = H_{tread-road} A_{cp} (T_{tread} - T_{road}) \tag{8.34}$$

In practice, contact heat transfer is typically influenced by contact pressure. In addition, the heat transfer rate may differ in the sliding and non-sliding regions. These factors are neglected here.

8.3.6.4 Heat Transfer with Inflation Gas

We model the inflation gas as having only thermal contact with the carcass as described by equation 8.35.

$$Q_{carcass->gas} = H_{carcass-gas} (T_{carcass} - T_{gas}) \tag{8.35}$$

8.3.7 Differential Equations

The rates of change of temperature of the three bodies are summarised in equations 8.36,8.37 and 8.38.

$$\dot{T}_{tread} = \frac{Q_{sliding} - Q_{tread->road} + Q_{carcass->tread} - Q_{tread->ambient}}{S_{tread} M_{tread}} \tag{8.36}$$

$$\dot{T}_{carcass} = \frac{Q_{damping} - Q_{carcass->tread} - Q_{carcass->ambient} - Q_{carcass->gas}}{S_{bulk} M_{bulk}} \tag{8.37}$$

$$\dot{T}_{gas} = \frac{Q_{carcass->gas}}{S_{gas} M_{gas}} \tag{8.38}$$

The values $S_{tread}, S_{carcass}, S_{gas}$ and $M_{tread}, M_{carcass}, M_{gas}$ are the specific heat capacities and masses of the three components.

8.3.8 Computational Procedure

The computational procedure for the tyre model is summarised below. The procedure computes the tyre forces and the rates of change of temperature of the bodies. The control inputs consist of:

1. Vertical Load
2. Road Speed
3. Slip Angle
4. Slip Ratio

Computation proceeds as follows:

1. Update inflation gas pressure from known gas temperature
2. Compute contact patch dimensions using vertical load and inflation gas pressure
3. Compute static and dynamic coefficient of friction based on contact patch pressure, tread temperature and sliding velocity.
4. Compute the distance x_t along the contact patch where sliding begins.
5. Compute Tyre Lateral and Longitudinal forces
6. Compute Heat Generation in tread due to sliding
7. Compute Heat Generation in carcass due to force generation
8. Compute Heat transfer rates between bodies
9. Compute Rates of change of temperature of bodies

Now all rates of change have been calculated, A differential equation solver is used to update the temperatures of the system elements.

8.3.9 Summary of model State, Inputs and Parameters

The following tables summarise the tyre state and control variables and give values for the tread compound and thermal parameters.

Symbol	Description	Units
T_{tread}	Tread Temperature	$°C$
T_{gas}	Inflation Gas Temperature	$°C$
$T_{carcass}$	Carcass Temperature	$°C$

Table 8.1: Tyre State Variables

Symbol	Description	Units	Value Front	Value Rear
μ_{REF}	Reference Static friction	-	2.25	2.25
μ_{base}	Lower limit of dynamic friction	-	0.7	0.7
μ_{peak}	Upper limit of dynamic friction	-	2.25	2.25
K_{shape}	Master Curve Shape Factor	-	0.615	0.615
K_{shift}	Master Curve Temperature Shift Factor	-	0.02	0.02
T_{REF}	Master Curve Reference Temperature	°C	45	45
G_{base}	Limit Compound Shear Modulus at high temperature	kPa	0.5	0.5
G_{TA}	Shear Modulus at temperature T_{GA}	kPa	2.5	2.5
G_{TB}	Shear Modulus at temperature T_{GB}	kPa	0.8	0.8
T_{GA}	Shear Modulus Reference temperature A	$°C$	25	25
T_{GB}	Shear Modulus Reference temperature B	$°C$	75	75
h_{tread}	Tread height	mm	5	5
K_{pcpp}	Friction roll off factor with contact patch pressure	0.1	0.1	
K_{refcpp}	Reference contact patch pressure	kPa	100	100

Table 8.2: Compound Data

Symbol	Description	Units	Value Front	Value Rear
V_g	Volume of inflation gas	m^3	0.0748	0.0825
E_z	Carcass Vertical Force Efficiency factor	-	0.03	0.03
E_x	Carcass Longitudinal Force Efficiency factor	-	0.03	0.03
E_y	Carcass Lateral Force Efficiency factor	-	0.03	0.03
$H_{tread->road}$	Heat transfer coefficient Tread to Road	$\frac{kW}{m^2K}$	12	12
$H_{tread->amb}$	Heat transfer coefficient Tread to Ambient (equation 8.32)	$\frac{W}{K}$		
$H_{carcass->amb}$	Heat transfer coefficient carcass to ambient	$\frac{W}{K}$	30	30
$H_{carcass->gas}$	Heat transfer coefficient carcass to inflation gas	$\frac{W}{K}$	8	8
$S_{carcass}$	Specific heat capacity of Carcass	$\frac{kJ}{kgK}$	1.8	1.8
S_{tread}	Specific heat capacity of tread	$\frac{kJ}{kgK}$	1.8	1.8
S_{gas}	Specific heat capacity of inflation gas	$\frac{kJ}{kgK}$	1.042	1.042
$M_{carcass}$	Mass of Carcass	kg	9.5	11.5
M_{tread}	Mass of tread	kg	0.5	0.5
M_{gas}	Mass of inflation gas	kg	-	-

Table 8.3: Thermal Data

Symbol	Description	Units

T_{road}	Track surface Temperature	$°C$
T_{amb}	Track air Temperature	$°C$
F_z	Vertical Load	N
α	Slip angle	-
κ	Slip ratio	-
v_x	Longitudinal car Speed of	m/s
v_s	Modulus of vector sliding speed at contact patch	m/s

Table 8.4: Tyre Control Inputs

8.4 Example Tyre Curves

Figure 8.9 shows an example longitudinal μ/slip curve for a front tyre with a vehicle speed of 100km/h, tyre pressure 1.1 bar at 90°C, and tread temperature 70°C. The contribution of sliding and non-sliding contact patch regions is shown. In the 'linear' low slip region of operation, the behaviour is dominated by the non-sliding region. As slip increases, more of the contact patch enters the sliding region and the dynamic friction force dominates. The μ/slip curves produced by the model are typical of the Pacejka data provided by tyre manufacturers. Figure 8.10 gives μ/slip curves across a range of tread temperatures. At

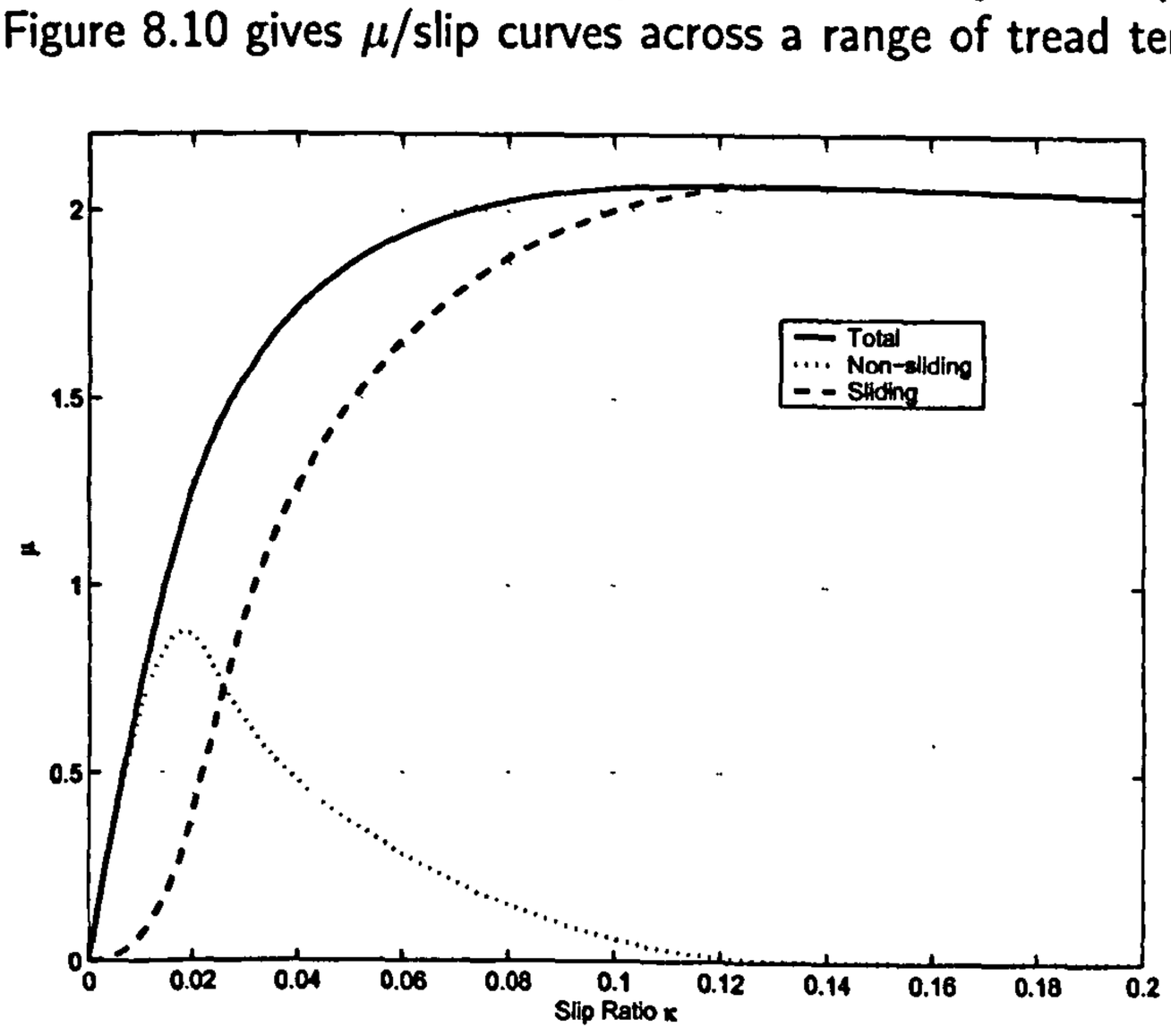


Figure 8.9: μ / Slip contribution from sliding and non sliding regions of contact patch

low temperature the following apply:

- The shear modulus of the compound is higher resulting in an increase in slip stiffness in the linear region.

- The dynamic μ rolls off more rapidly with sliding speed resulting in a decrease in grip at high slips.

The colder tyre is hence stiffer and less forgiving when overslipped.

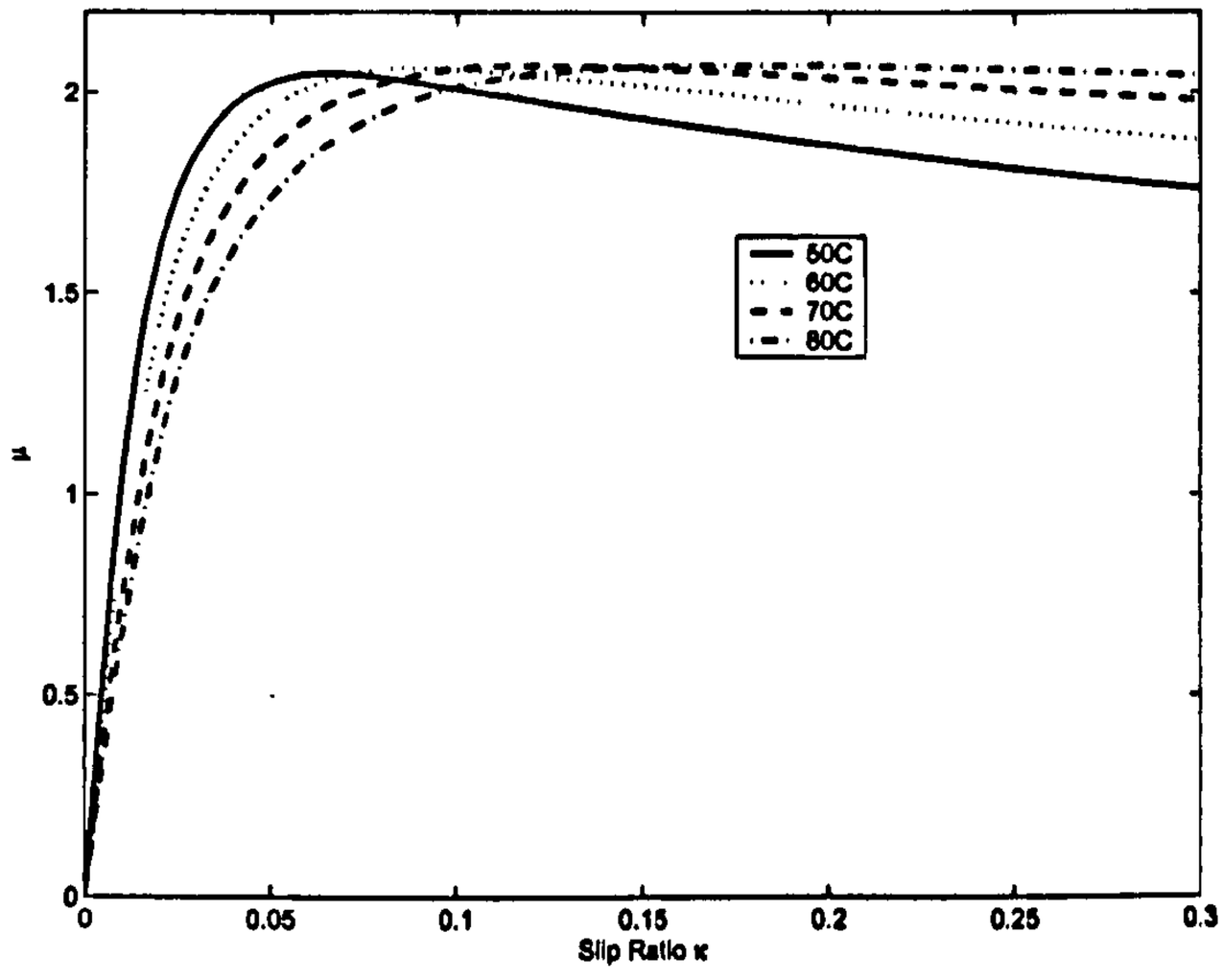


Figure 8.10: μ / Slip ratio curves across a range of temperatures

8.5 Simulation Results

The results described below use the QSS car model with set-up A. We begin below by simulating the effect of manoeuvre time on two short manoeuvres using fixed tread temperatures and with the thermodynamic aspects of the model disabled. We then move on to simulate dynamic temperature effects using short manoeuvres and then a longer manoeuvre involving multiple laps of the Silverstone National circuit.

8.5.1 Analysis with fixed tyre temperatures

Firstly we establish some baseline manoeuvre time results with the tyre temperatures and pressures fixed at constant values. This confirms the ideal operating temperature range of the tread. The tyre pressures and gas temperatures are configured as: front 1.15 bar, rear 1.05 bar (at 90° gas temperature). All four tyres are set to the same temperature. We simulate the single 90° corner and the Becketts manoeuvre. Figure 8.11 shows the manoeuvre time for tyre tread temperatures in the range 47-100°C for the single 90° corner. Below 55°C the car is unstable and this is reflected in the scatter of the manoeuvre times. The temperature for minimum manoeuvre time is approximately 85°C. Figure 8.12 shows similar data for the Becketts Manoeuvre. Figure 8.13 shows the steer angles, yaw stiffness and control moment derivatives for the 65° and 80° scenarios in the single 90° corner case. The colder tyre case shows less N_β in braking and more in mid-corner, indicating less entry

stability and more mid-corner stability. The control moment derivative N_δ is higher for colder tyre in straight line running due to the higher slip stiffness. This also results in an increase of N_δ during the open-loop unstable phase of braking/entry. This will result in the car feeling more 'pointy' or 'nervous' to the driver on initial brake/turn in.

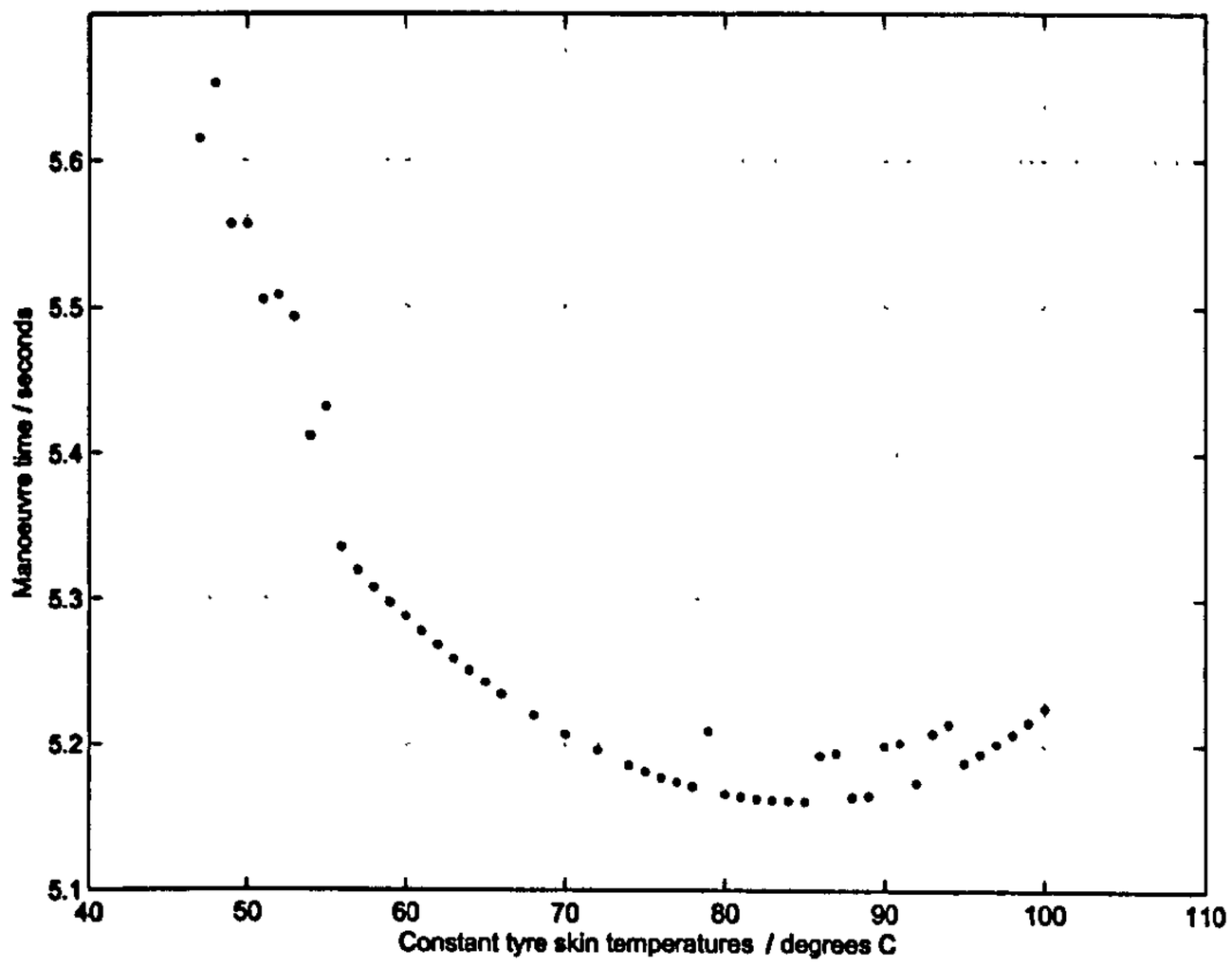


Figure 8.11: Righthander manoeuvre time effect at fixed tyre temperature

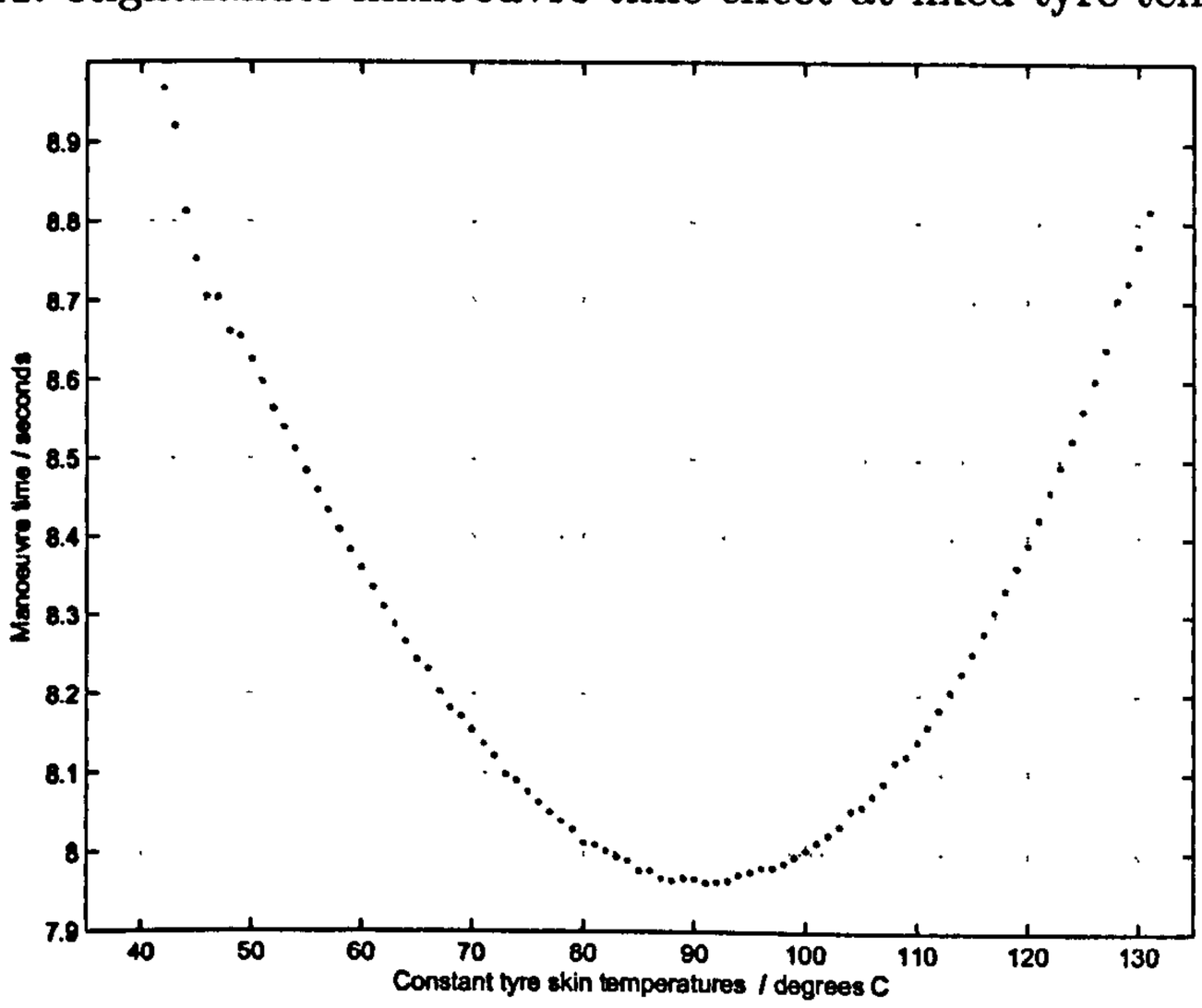


Figure 8.12: Becketts manoeuvre time effect at fixed tyre temperature

8.5.2 Track/Ambient Temperature Effect (short Manoeuvres)

Figures 8.14 and 8.15 show the effect of track temperature for the single 90° corner and Becketts manoeuvres with the tyre model thermodynamics enabled. The initial tyre states

are as follows: T_{gas}=90°,P_{gas}=1.1 bar front, 1.0 bar rear,T_{tread}=60°. These results are discussed later.

8.5.3 Track/Ambient Temperature Effect (longer Manoeuvre)

In this section we study the effect of track and ambient temperature over several laps of the Silverstone National Circuit (Figure 8.16). The resulting manoeuvre times are in the order of 4 to 5 minutes. This allows the slower thermodynamics of the carcass and gas to be considered. We consider track temperature in the range 15 to 35 °C. The ambient temperature is also modified such that $T_{amb} = T_{road} - 5$. Figure 8.18 shows the tread temperatures for all four tyres over the first three laps. The results are representative of those observed with tyre surface temperature sensors on real vehicles. Tread temperatures increase rapidly on braking / turn entry and decay in an exponential manner along the following straight. Figure 8.19 shows the heatflow along each path in the model. With the model calibration used, the dominant heatflows are carcass->tread and tread->road. The tyre heat generation processes are show for the left (loaded) rear and right (unloaded) front over a single lap in figure 8.20. The loaded rear tyre generates most of its heat due to deflection in the carcass. This tyre experiences little sliding due to the high vertical loads. The unloaded front generates significantly less heat overall with a higher proportion due to sliding. Due to nature of the circuit and vehicle set-up (forwards roll moment distribution) this tyre is unloaded through most of the corners. The exception being the left hand corner at the end of the club straight (Brooklands). The carcass and gas temperatures are shown in figure 8.21 over a seven lap run. The rear tyre carcass and gas increase from the initial temperature and stabilise to a lap average of around 107.5°C. The loaded front undergoes a small increase to 92.5°C. The unloaded front cools to approximately 77.5°C. The tread temperatures are shown in figure 8.22 over the seven lap outing. The carcass temperatures are reflected in the minimum skin temperatures seen through the lap. The lap minimum unloaded front temperature decreases over the first 4 laps. The rears increase over a similar period. The corresponding inflation gas pressures are shown in figure 8.23. Figures 8.24 and 8.25 show the inflation gas pressures over seven laps for the loaded rear and unloaded front tyres for track temperatures of 15, 20, 25 and 30°C. As expected the stabilised pressures increase in line with track temperature. Finally, we consider the effect of track temperature on manoeuvre time. It should be noted that we do not model the reduction in fuel load or any tyre wear effects. Figure 8.26 shows the evolution of lap time against lap number for four track/ambient temperatures. The lap times stabilise in line with the tyre pressures as should be expected. Figure 8.27 shows the lap times against track temperature for the various lap numbers. For each lap number, there is a well defined track temperature for minimum lap time.

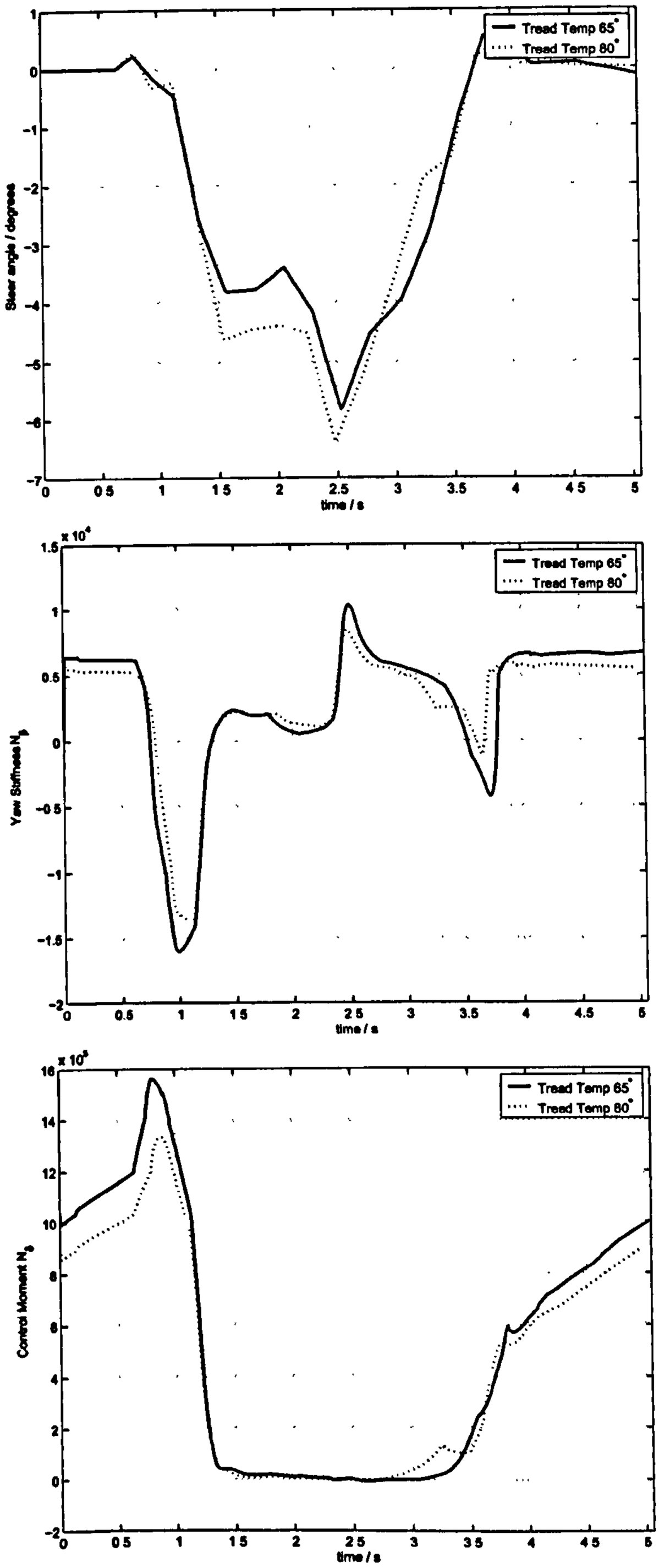


Figure 8.13: δ,N_β,N_δ with fixed tread temperatures of 65 and 80 °C

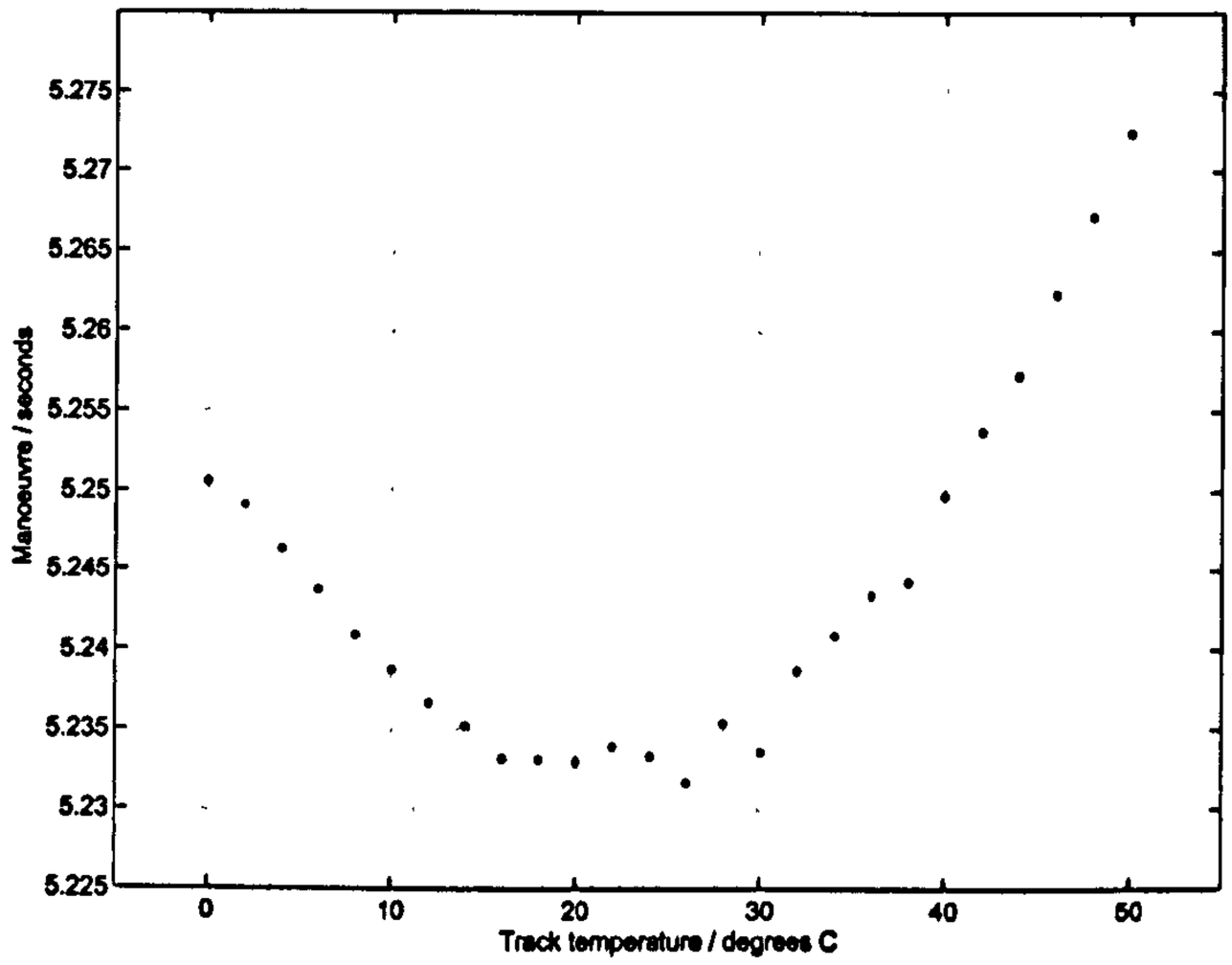


Figure 8.14: Righthander manoeuvre time effect of track temperature

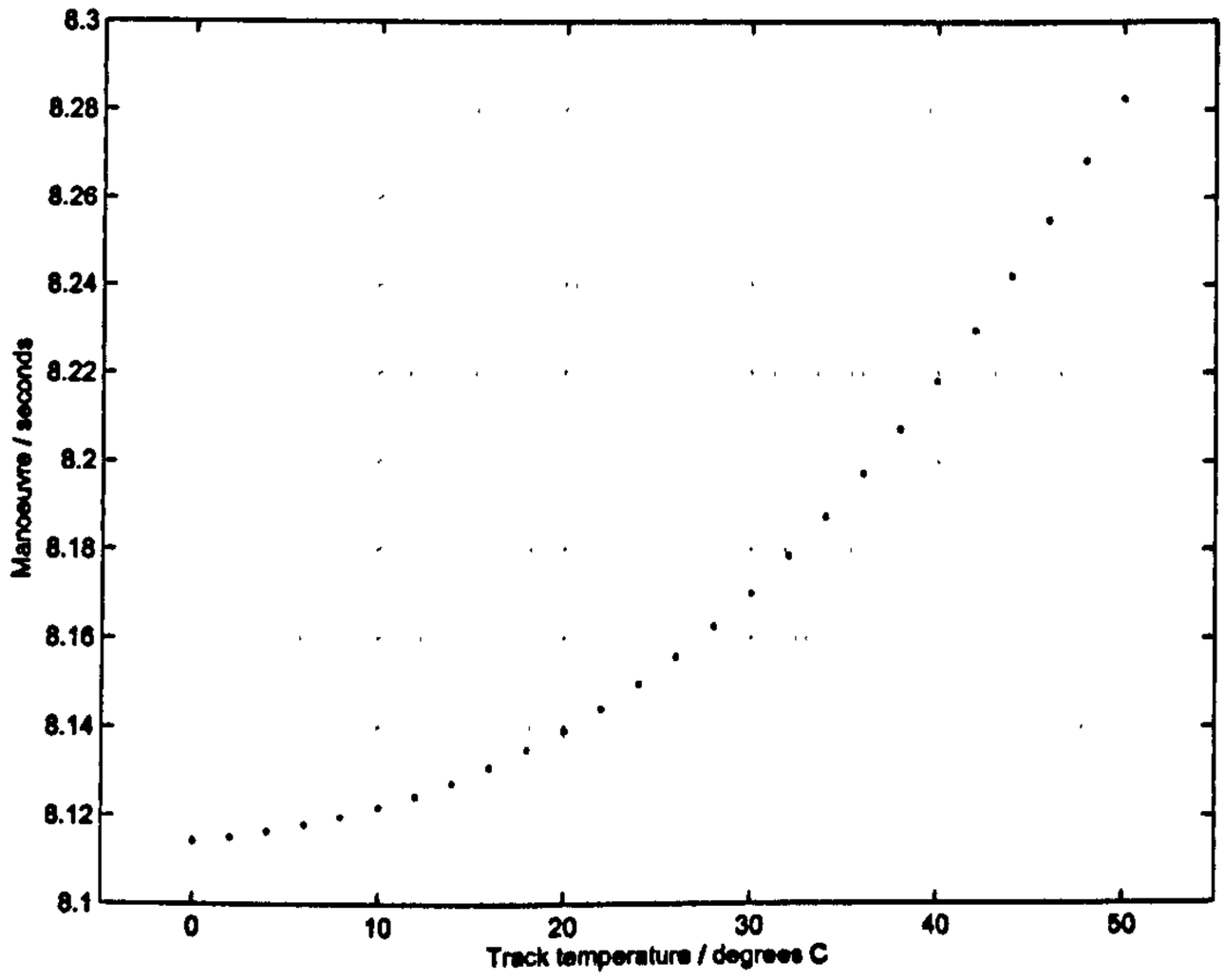


Figure 8.15: Becketts manoeuvre time effect of track temperature

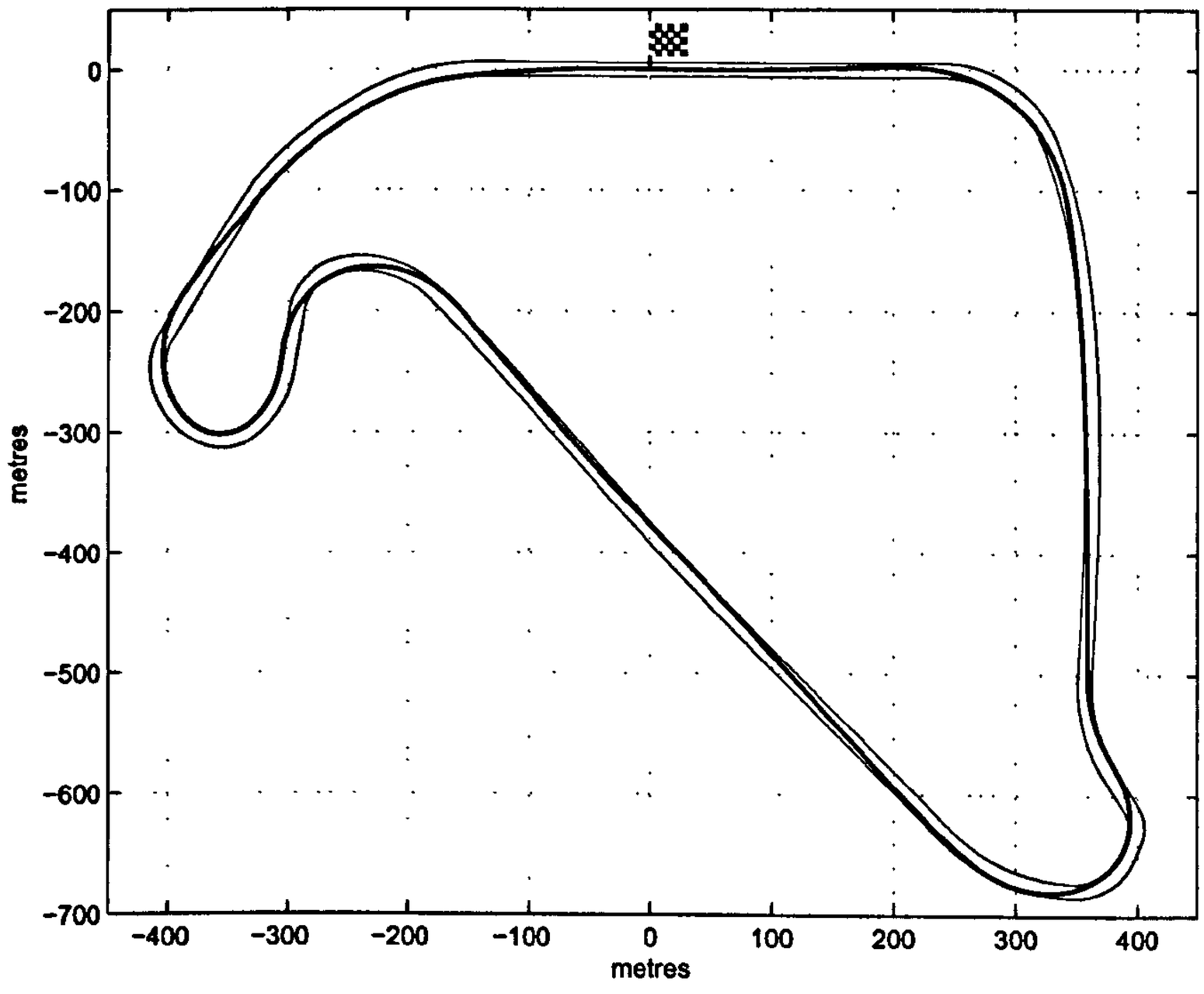


Figure 8.16: Silverstone National Circuit

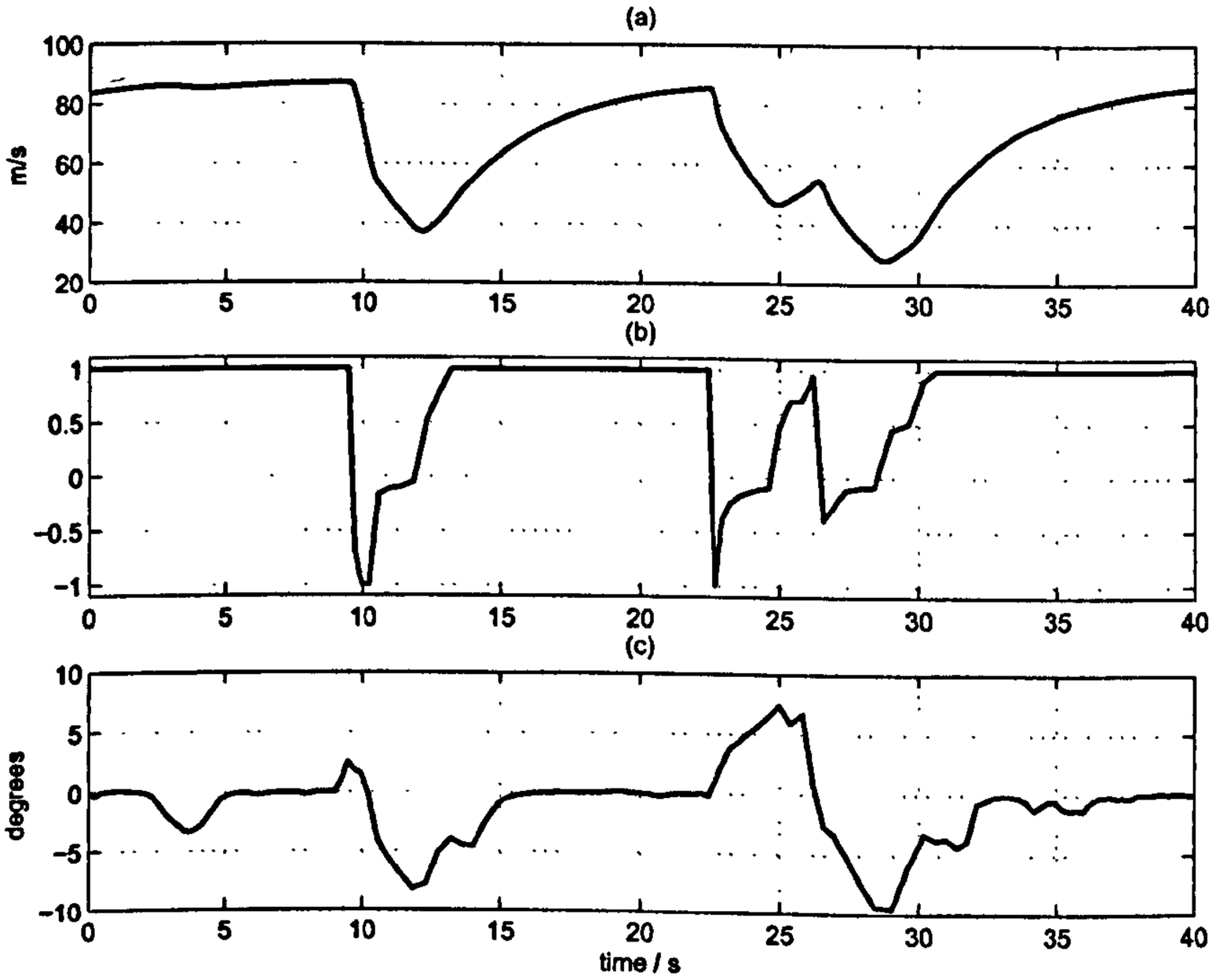


Figure 8.17: (a) Speed (b) Throttle/Brake (c) Steer

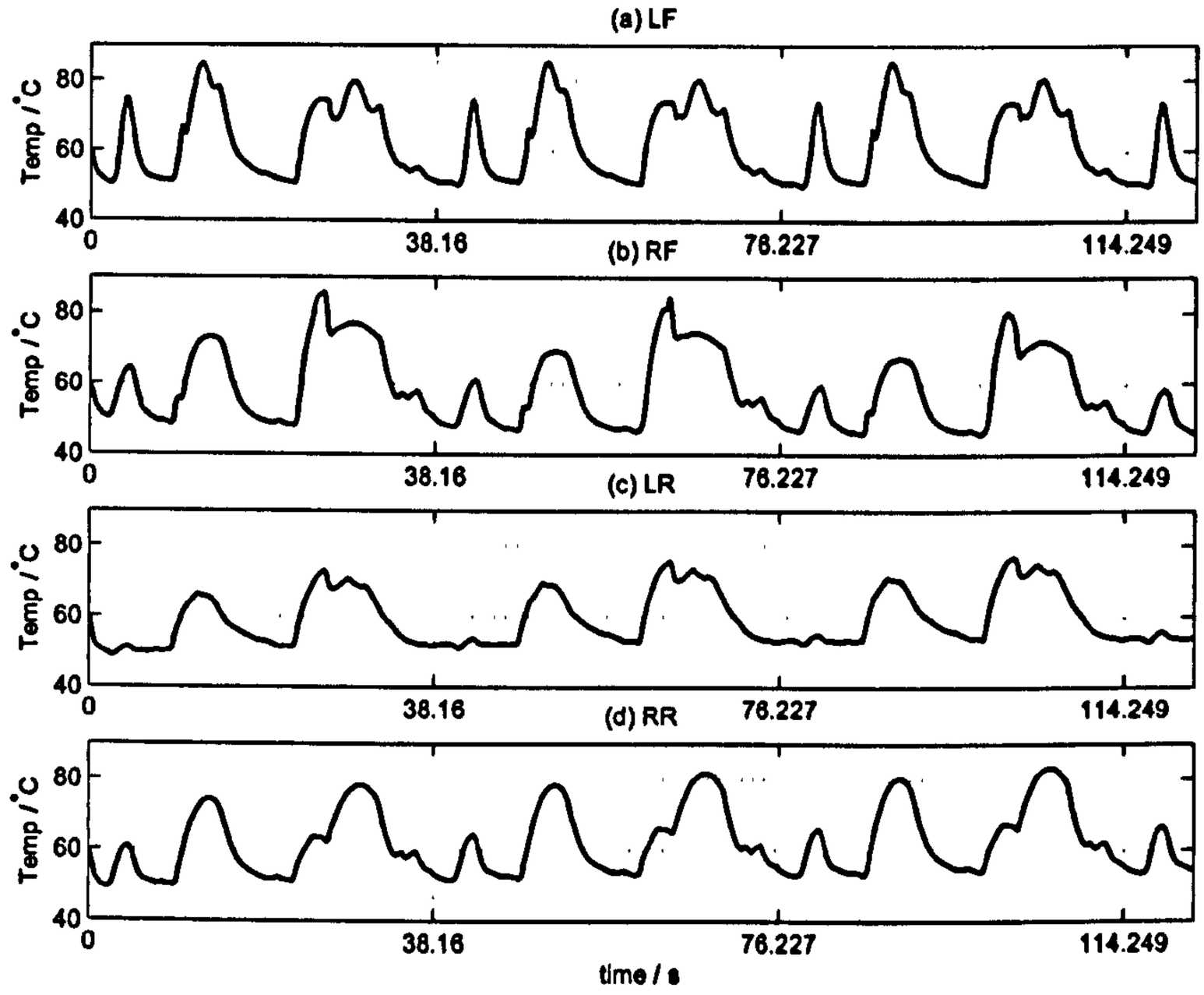


Figure 8.18: Tread Temperatures over first 3 laps. Track 30°, Ambient 25°

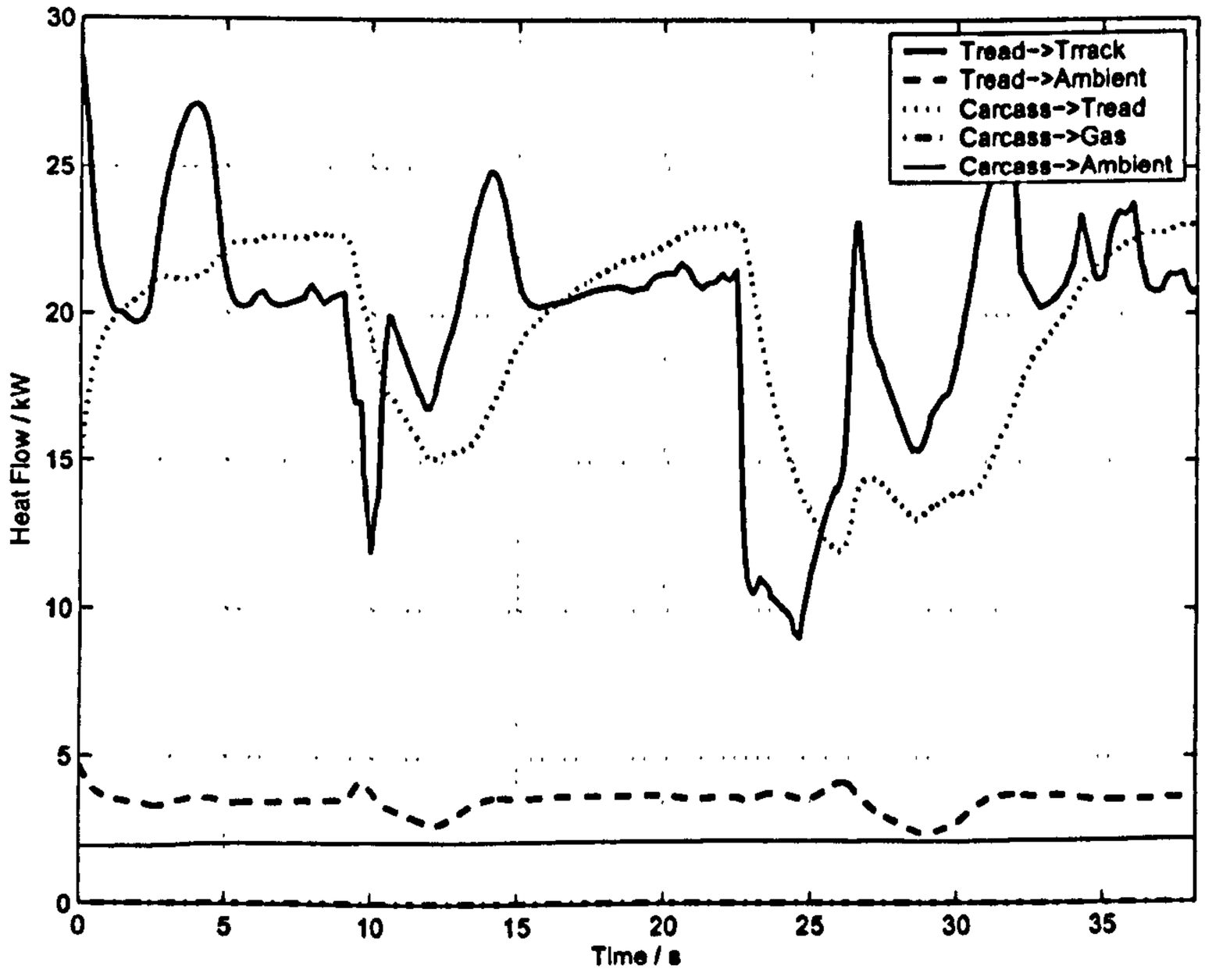


Figure 8.19: Heatflows in Left Rear Tyre

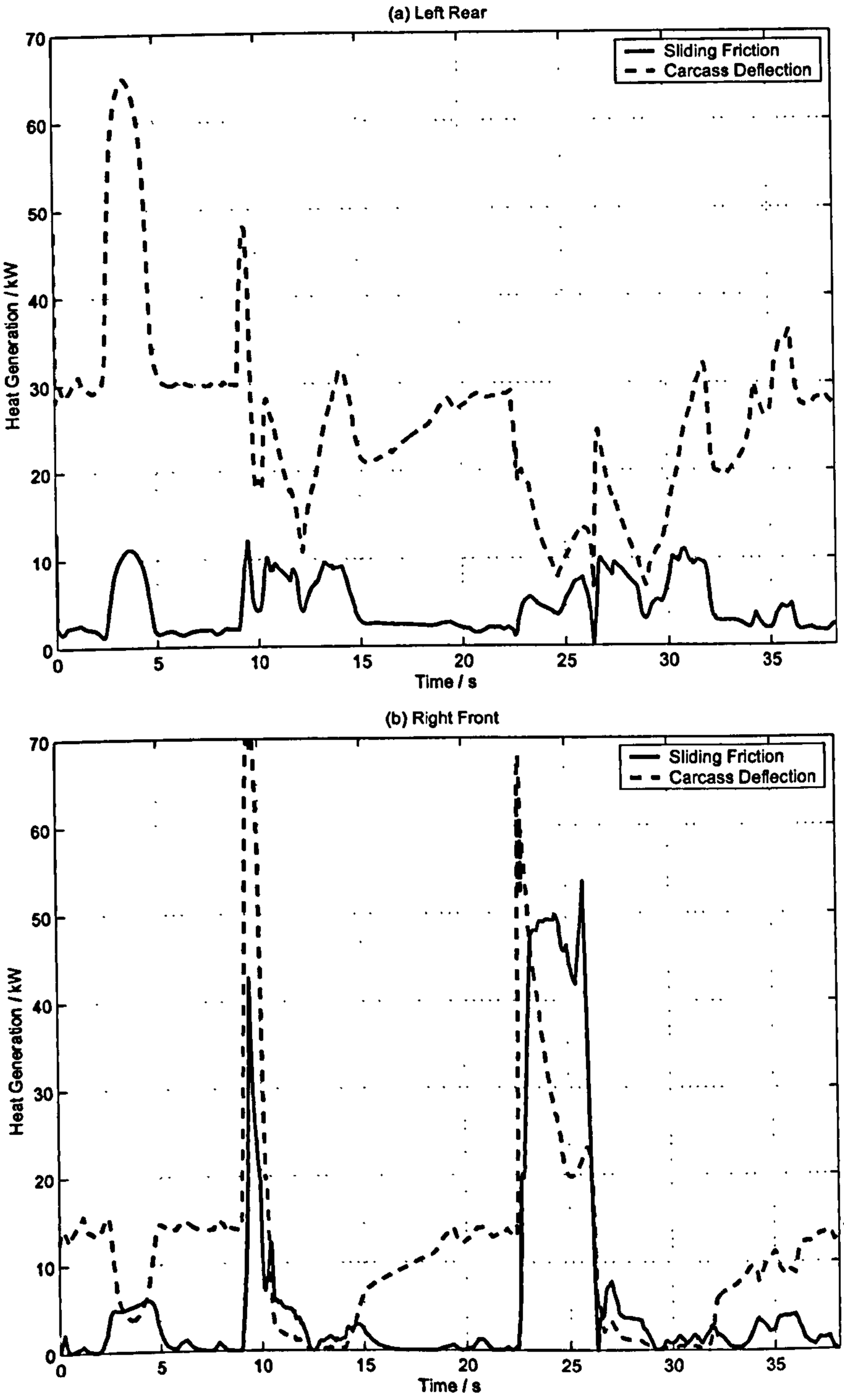


Figure 8.20: Heat Generation in Left Rear and Right Front Tyres

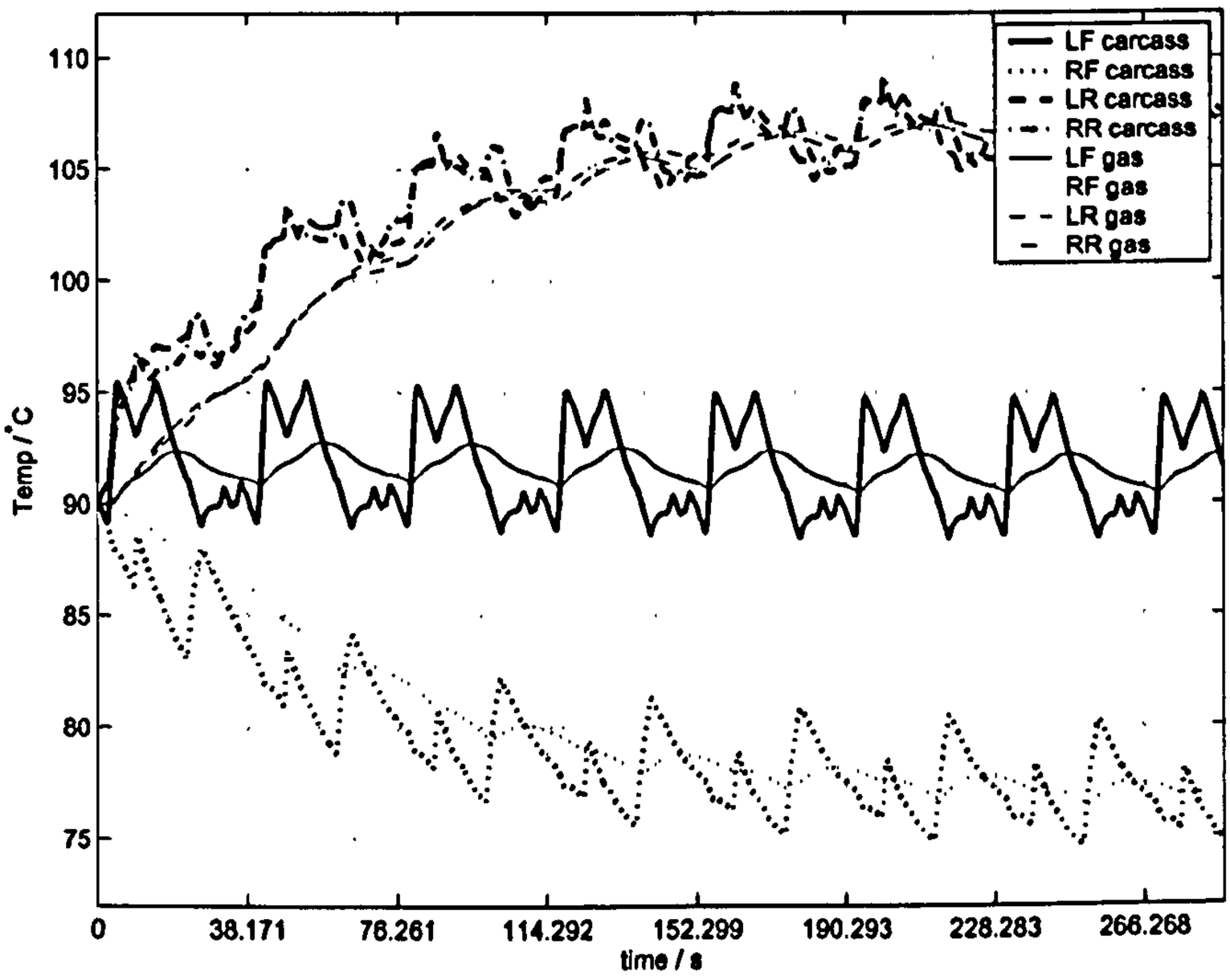


Figure 8.21: Carcass/Gas temperature evolution over 7 laps. Track 30°, Ambient 25°

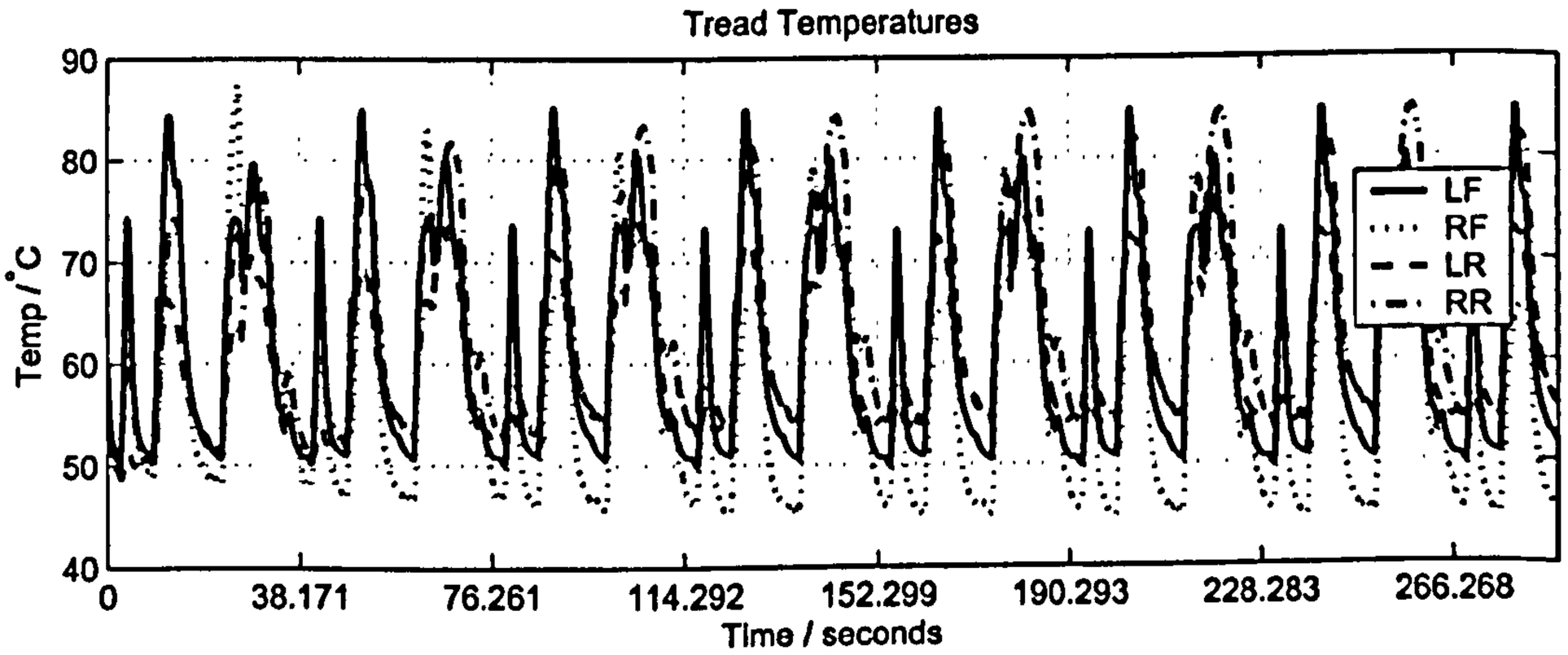


Figure 8.22: Tread temperature evolution over 7 laps. Track 30°, Ambient 25°

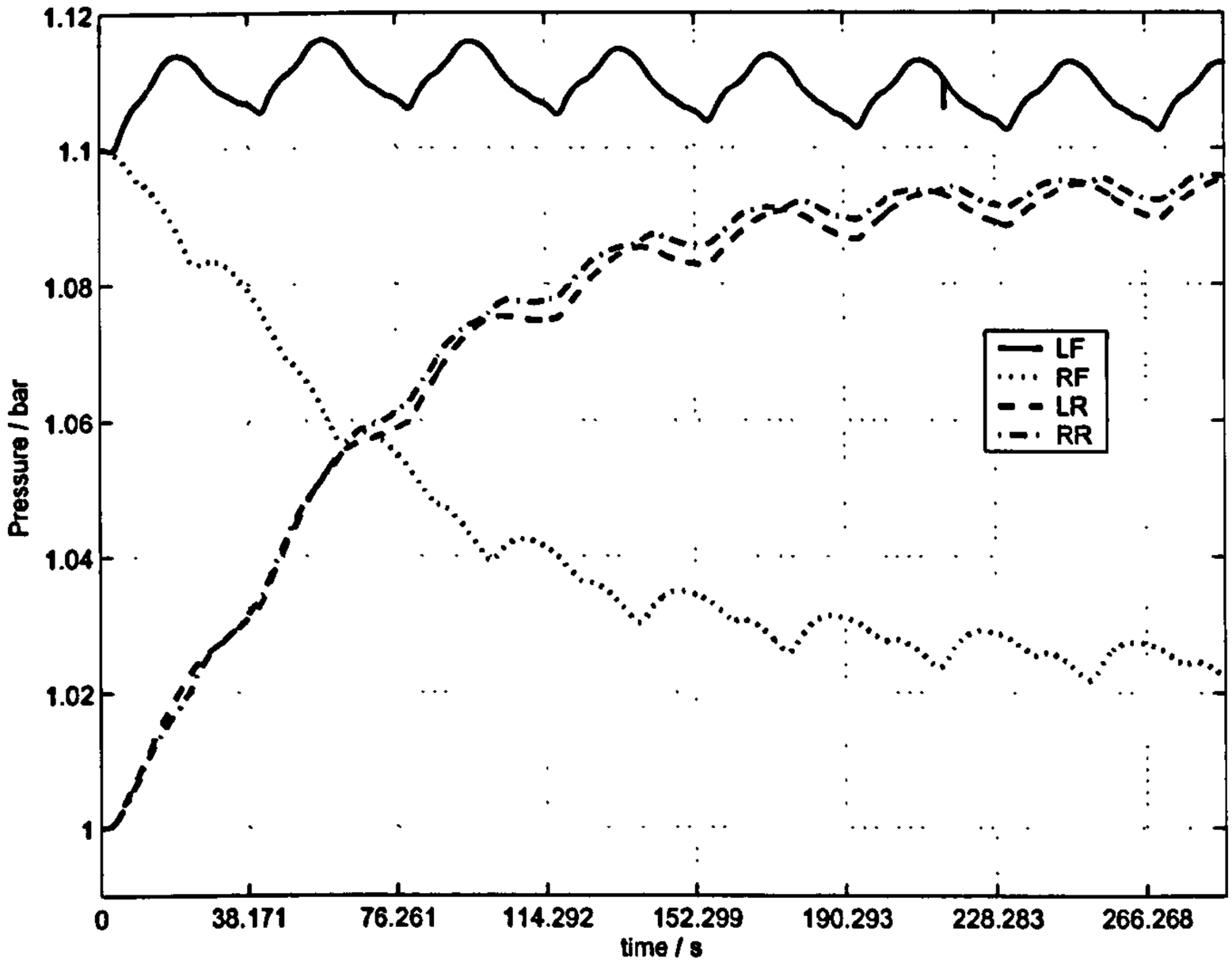


Figure 8.23: Tyre Pressure evolution over 7 laps. Track 30°, Ambient 25°

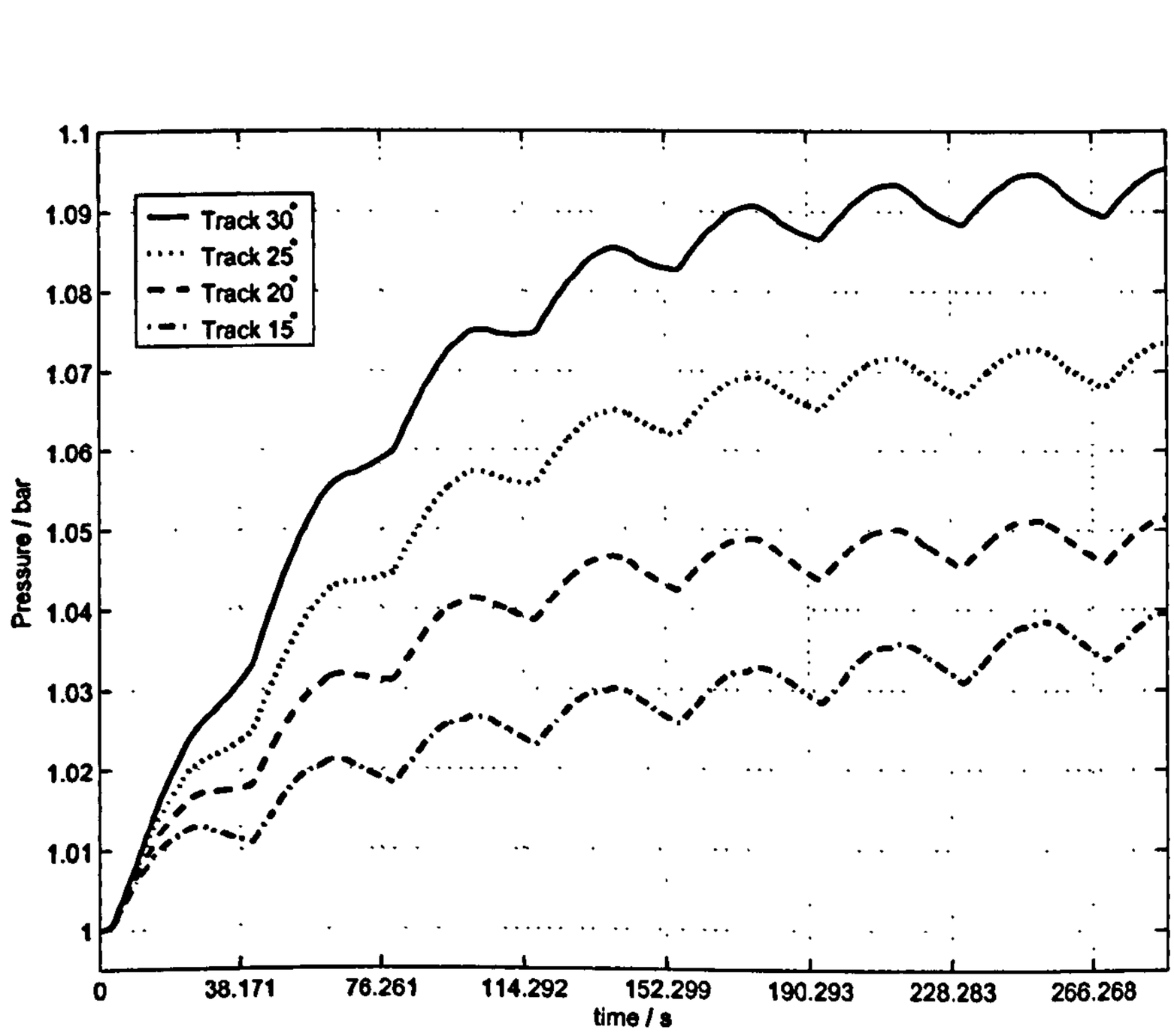


Figure 8.24: Left Rear Tyre Pressure evolution over 7 laps

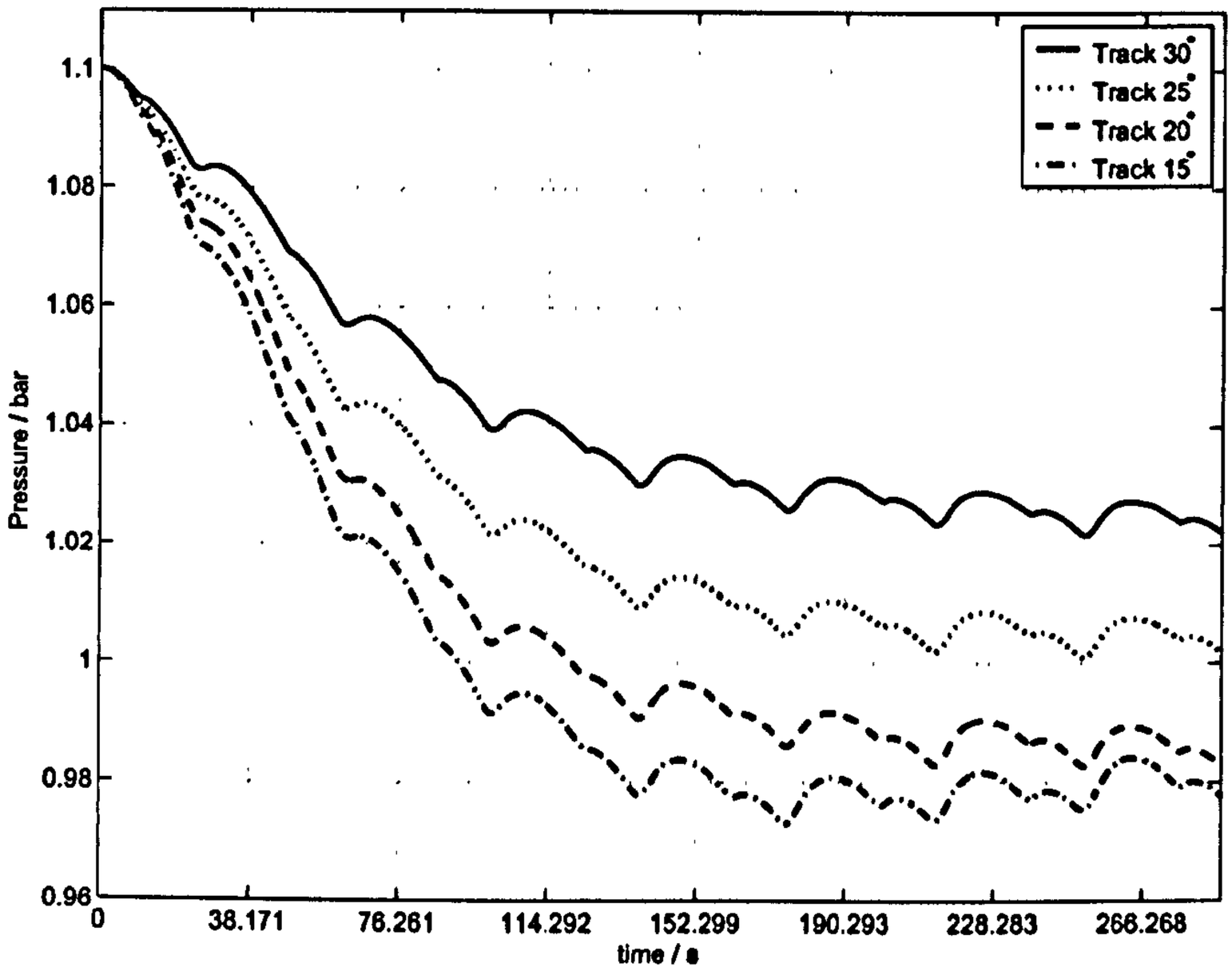


Figure 8.25: Right Front Tyre Pressure evolution over 7 laps

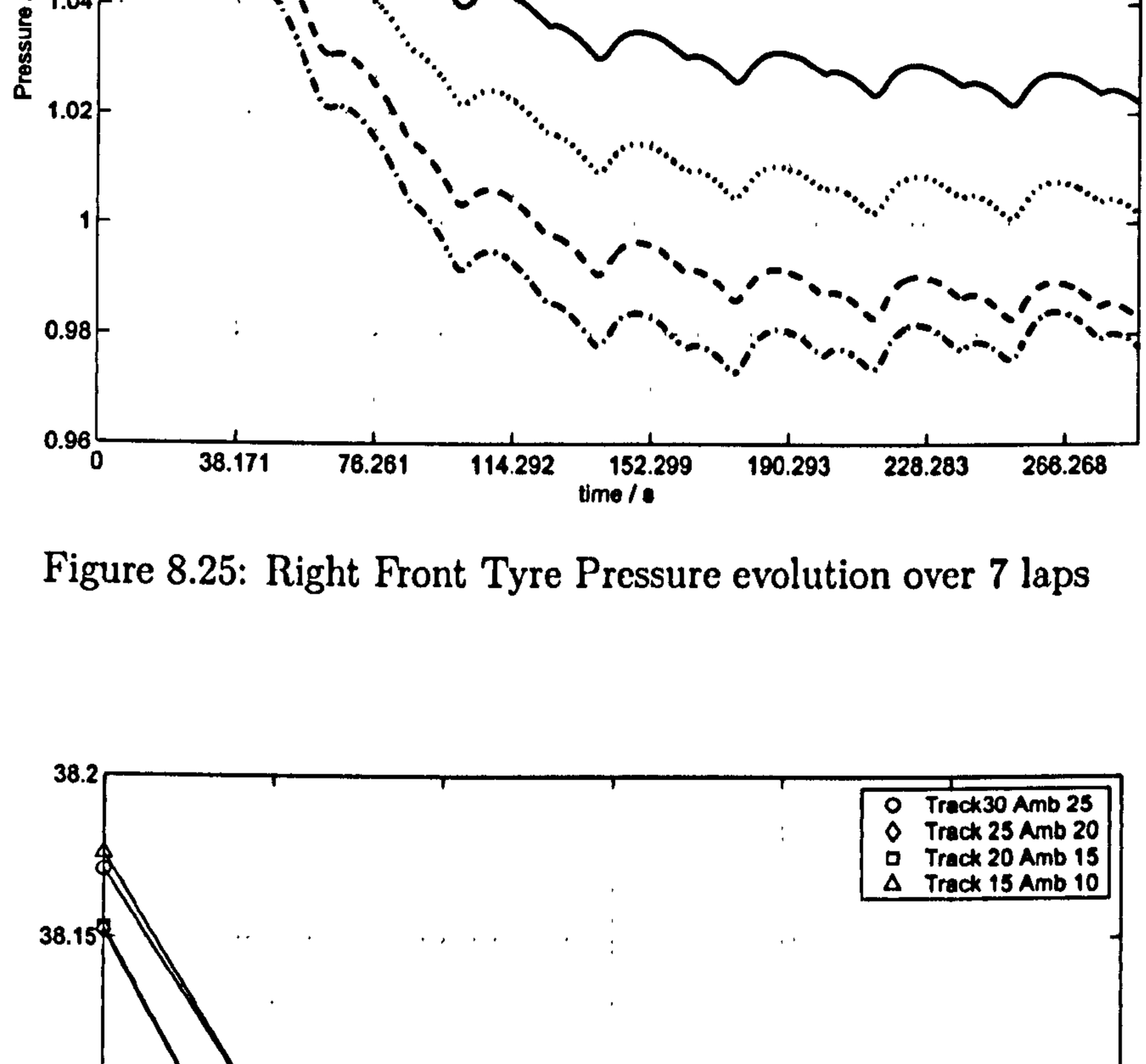


Figure 8.26: Lap time evolution over 7 laps

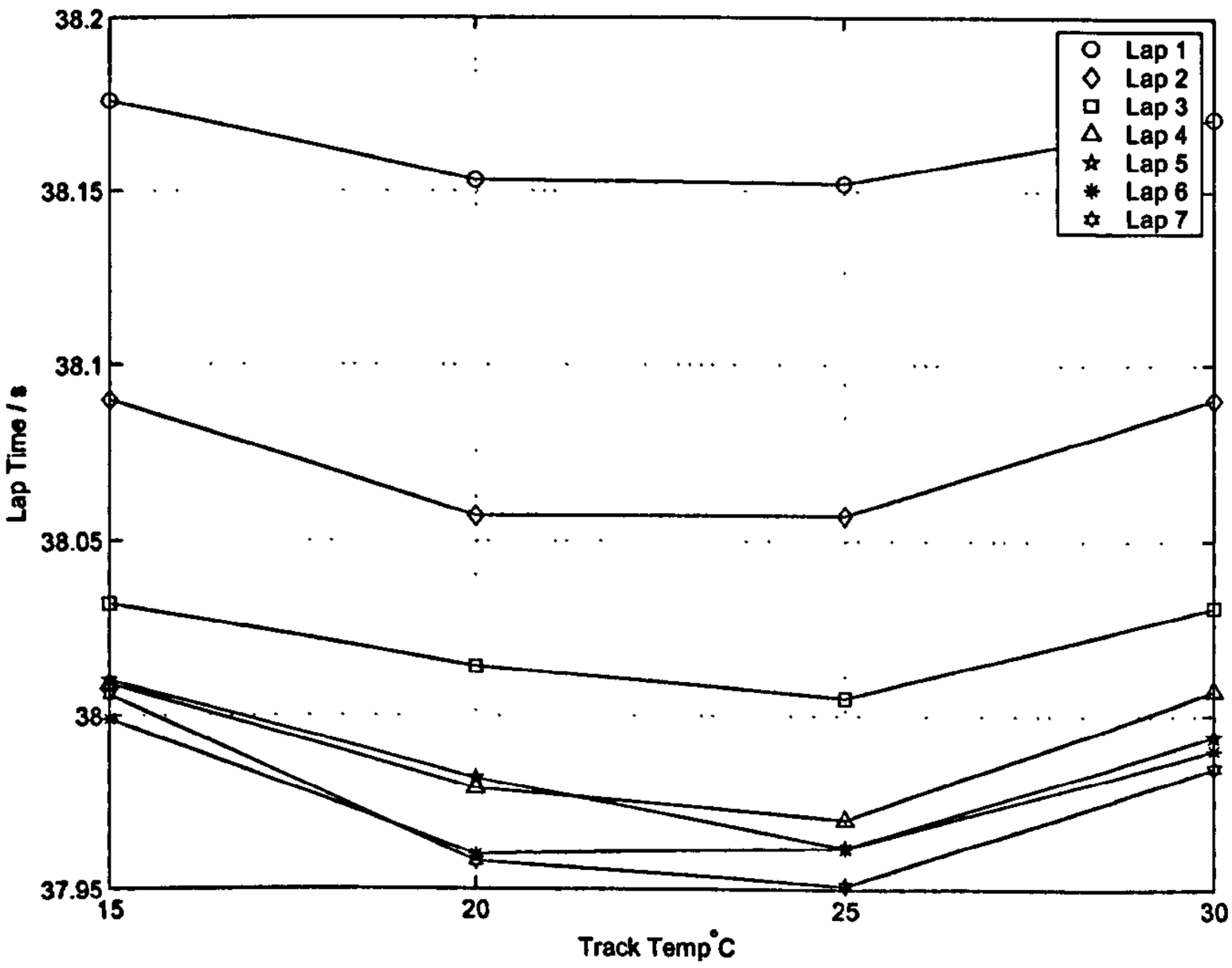


Figure 8.27: Lap time influence of track temperature and lap number

8.6 Discussion

The introduction of a thermal tyre model does not appear to cause any particular problems for the presented MTM method. Results remain consistent and all parametric experiments show well defined minima. The developed tyre model is quite simple but is able to demonstrate certain aspects of the behaviour of real racing tyres such as an ideal operating window in both tyre and track temperatures.

8.6.1 Model Behaviour

The thermal dynamics of the tread and carcass operate on quite different timescales. In the calibration used here the thermal inertia of the carcass is approximately 20 times that of the tread. The calibration also results in the main way that heat leaves the tyre being through the contact patch into the road surface. The tread is in direct contact with this principal heatsink and tends to cool rapidly as the tyre sliding velocities fall away on corner exit. When running with little slip (for example the front tyres on a straight section of track) heat generation in the tread is minimal. Under these circumstances, the temperature of the tread is determined (typically) by heat flow into the tread from the carcass and heatflow out of the tread into the track. The carcass can be considered as the tyres heat store, it prevents the tread temperature from falling to the track temperature in this situation. When arriving at a corner following a straight, the tread temperatures will typically be lower than ideal but will rise rapidly with the onset of slip. In our calibration tread temperatures are typically lower than carcass temperatures hence heat is almost always flowing out of the carcass into the tread. The tread hence acts an insulating blanket between the carcass and road surface. Any general increase in tread temperature will help retain heat in the carcass which will in turn help maintain tread temperature in low slip running. As previously mentioned, in real racing tyres tread temperatures may be significantly higher in the contact patch than in the general tread. Hence in the real tyre, heat may routinely flow from the contact patch into the carcass.

The short manoeuvres prove useful for studying the effects of the rapid thermodynamics of the tread. For these manoeuvres, the bulk and gas temperatures are effectively constant. The track/ambient temperature against manoeuvre time plots (figures 8.14 and 8.15 each show well defined minima. With the initial conditions used (Carcass 90° and tread 60°), the lower speed (single 90°) corner prefers a hotter track in order to get the tyres at temperatures suitable for the sliding velocities associated with the manoeuvre. The optimal track temperatures are in the range 15-30°. The faster (Becketts) manoeuvre prefers a cooler track surface. In this manoeuvre the tyre tread and carcass heating are higher (through higher loads and forces) and this appears to push the tread temperature beyond optimal values. Hence a lower track temperature helps to bring them back into range. This phenomenon may be due to a 'thermal runaway' feature of the model and is discussed later.

It is also worth noting that the tyre temperatures are representative of the perfect driver (or traction control / ABS augmented human driver). The real driver is more likely to overslip the rears and also more likely to have less sliding velocity on fronts to prevent graining (discussed later).

8.6.2 Sliding Speeds and Operating Window

The racing tyre must be designed taking into account the range of sliding velocities it will see on the racetrack. Considering the pure cornering case (no longitudinal forces on the tyres). For any particular racetrack, with a given driven line and speed profile, the yaw rate contribution to the lateral velocity seen at the rear axle is dictated by the vehicle wheelbase and longitudinal centre of gravity location. It will also be affected by the chassis lateral velocity but at most by a factor of two. Hence the sliding velocity range is dictated by predetermined factors. Since the lateral and longitudinal performance of a tyre are similar, the same applies for longitudinal sliding velocities. As previously stated, rubber friction is determined by excitation frequency rather than sliding speed. The excitation frequency is determined by the sliding speed and the roughness of the track surface. Hence this is also a key factor affecting the design of a tyre to suit a particular circuit. The sliding speed at which μ_d is maximised is determined by the tread temperature. Hence the tyre must be tuned such that it achieves an operating temperature to maximise grip at the desired sliding velocities. In our model, we can control the basic operating temperature of the tyre by adjusting the losses in the carcass when generating forces. Further consideration must be given when designing a front and rear tyre that the vehicle handling remains safe across a range of typical operating track and ambient temperatures. If under any circumstances, the front tyres produce significantly more grip than the rears, the vehicle handling may move into a dangerous oversteer regime. Where a compromise must be found, one imagines that the tyre designer must err on the side of caution and produce a front / rear balance that tends towards understeer when operating outside the ideal track temperature window by achieving a suitable front/rear temperature balance. In our model we can tune the operating window of the tread compound to a particular manoeuvre using the parameters T_{REF} and K_{shift}. T_{REF} sets the temperature at which μ_d achieves a maximum at 1 m/s sliding speed. K_{shift} determines the rate at which the peak μ_d sliding speed changes with temperature. One presumes, nature being what it is, that the tyre compounder has a choice between a high grip level across a narrow temperature range or a lower grip level across a wider working range. Since we know that the tread temperature and sliding speeds change rapidly through a manoeuvre, optimal values for T_{REF} and K_{shift} are not obvious. It would be interesting to modify the MTM method by adding tyre parameters as independent variables. The following approaches may prove interesting:

- Add T_{REF},K_{shift} and the shear modulus parameters as independent variables to the optimizer.

- Disable model thermodynamics and specify tyre tread temperatures at specific times/distances through a manoeuvre in a similar manner to the throttle/steer inputs. This would establish ideal tread temperature operating profiles.

8.6.3 Graining and Wear

In the model presented, we have assumed that arbitrarily high sliding speeds can be applied to the rubber. This allows the tread to be easily heated from any initial temperature simply by applying sufficient load and sliding speed. In the case of the front tyre where the steer element of the front slip angle is under control of the optimizer it is always possible to warm

the fronts simply by steering. In practice, the following factors become important at high sliding velocities and low tyre temperatures:

1. The sliding friction may become 'stick-slip'. This results in the tyres skipping or 'chattering' across the road surface resulting in a loss of grip.

2. If a soft rubber at low temperature slides over a hard substrate at sufficiently high speed, the rubber may crack/tear and pieces of rubber may break away and roll under the sliding contact interfering with the surfaces in contact and disrupting the processes by which friction is generated.

The second of the above is usually termed 'graining'. The graining process can be demonstrated by rapidly sliding a soft pencil eraser over a hard surface. The eraser will tear and rolls of soft rubber will separate. In the case of a tyre small balls of rubber are formed which gather together and disrupt the tyre/road contact. Graining occurs because the sliding velocity is too high at the operating value of the tread rubber. Persson [42] indicates the relationship between friction and wear given in figure 8.28. Here we see that there is a double penalty to operating outside of the optimal sliding speed range. Not only is grip reduced but wear is increased. Graining can be considered as an extreme case of wear. Our simple model for sliding friction and the associated heating does not take into account

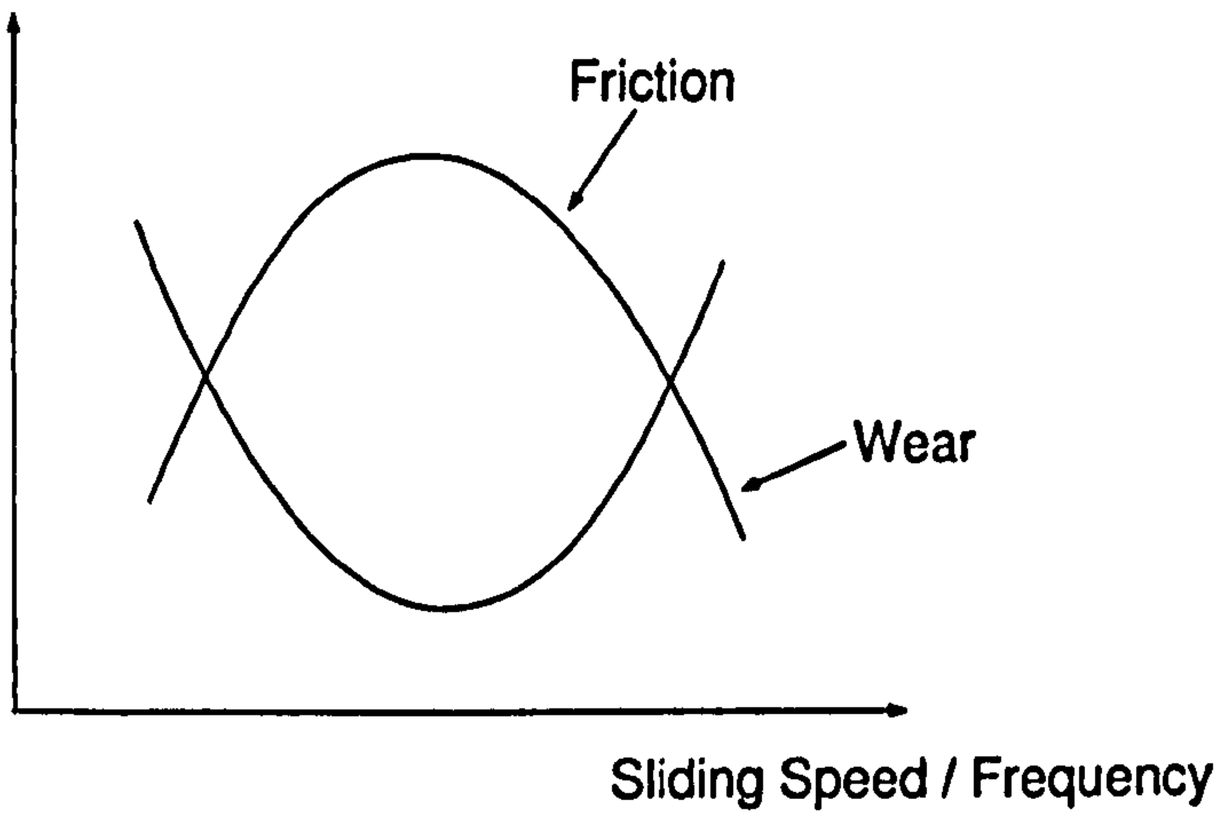


Figure 8.28: Rubber Friction and Wear

such wear factors. In the presence of cold tyre induced understeer,the MTM method can apply an arbitrary sliding velocity to the front tyres simply by applying sufficient steer angle. This can result in a 'thermal runaway' scenario: Applying a high steer angle raises sliding speed and hence tread temperature, the hotter rubber achieves its peak μ_d at higher sliding velocities which then requires the optimizer to apply more steer angle to achieve optimum grip. Hence the temperature increases further and the process re-inforces itself. In order to prevent this scenario, a 'graining' constraint could be added to the MTM method that restricts the acceptable range of sliding speeds based on tread temperature. Such a constraint could be used to develop an anti-graining tyre warm-up procedure.

Chapter 9

Conclusions

You accept the fact that Formula 1 is totally futile and stupid and still you carry on...

Sir Jackie Stewart

The preceding chapters have developed a robust approach to transient manoeuvre time simulation capable of operating with complex non-linear vehicle and tyre models. In this chapter we review the findings, discuss the strengths and weaknesses of the approach used and make recommendations for further work in this area.

9.1 Discussion of the MTM problem

We have shown that the MTM problem as formulated here has non-smooth objective and constraint functions. This is most probably due to the nature of the tyre. When operating in the linear region of the tyres, engine torque can be used to control tyre slip in a linear fashion. However, when operating at the limit of tyre performance, a very small increase in engine torque will result in a large rapid change in slip. This longitudinal super-saturation of the tyre also rapidly reduces its lateral capability causing a sudden change in yaw moment. Since the MTM method seeks solutions at the limit of tyre performance we see evidence of highly non-linear vehicle behaviour close to time optimal solutions. We have also shown that this formulation results in the existence of well defined local minima. Many of the presented results show two or more distinct solutions. We cannot formally prove that a global minimum has been located but through performing repeated optimizations from many different values of the initial vehicle controls we can be confident that a global minimum point has been reached.

9.2 Discussion of the developed MTM Method

The most significant factor in the success of the MTM algorithm is the use of a feasible NLP method to remove the most non-linear and sub-optimal areas of the solution space from the problem. In the initial phase of the work a non-feasible code was used (SNOPT [21]) and most attempts at finding a solution resulting in the NLP code terminating due to the vehicle trajectory entering a highly understeer or oversteer regime. In addition to this, great care must be taken such that the definition of the functions defining the vehicle control inputs and constraints do not posses any discontinuities (however small). Such problems can be identified by careful study of the controls and constraints at the solution points as defined in section 2.12.2.

9.2.1 Validity of Results

We have demonstrated that the MTM method is capable of producing very accurate results and that repeatible convergence can be achieved from very different initial values for the vehicle controls. The fact that the same solutions can be found both with the time and Fourier bases enhances confidence that the minima found are related solely to the problem and are not artefacts of the MTM algorithm. The use of tyre efficiency diagrams and vehicle stability derivatives give a strong indication that the resulting vehicle state trajectories are optimal.

9.2.2 Computation Time

In the course of the research the author has pursued the goals or robustness and accuracy above all else. The main weakness of the developed method is its high demand for computational resources. If full accuracy is pursued (the NLP code is not limited in the number of major iterations) with random initial values for the vehicle controls, a full lap of the Jerez circuit with 10m control spacing and 250m preview will take approximately 8 hours of compute time using a single core of an Intel T7700 2.4GHz processor. Since most of the computer resources are consumed evaluating the objective function, this solution time scales with the complexity of the vehicle model. The efficiency of the finite horizon aspect of the algorithm is discussed later.

Plots showing objective value against iteration number (figures 3.6,3.7) indicate that progress can be very slow for several iterations and may then rapidly speed up. Progress at each major step may be as little as 1 μs. Typical optimal trajectories for racing cars are at the limit of stability, hence at the solution very small changes in the controls result in a sudden increase in manoeuvre time. At each major step, the NLP code evaluates the gradients at the current point and establishes the search direction to improve the manoeuvre time and maintain feasibility. When close to the stability limit of the vehicle it may not be possible to move very far in this search direction before the vehicle spins. Hence progress can be very slow. Typically a less stable car set-up (more tendency to oversteer) requires more iterations of the NLP code to converge. We consider some possible approaches to reducing the computer time to find a solution below:

9.2.2.1 Parallel Processing

Most of the time consumed by the method is used in evaluating the gradients of the objective and constraints using finite differences. This procedure could be very easily parallelized with independent variables being assigned to parallel processes. The LORA software (Appendix F) has been designed with parallelism in mind from the outset and the author plans to implement the code on a network of processors in the near future.

9.2.2.2 Closer to optimal Initial Control Values

A fairly uncompromising approach has been used in this work in seeding the initial control values as pseudo-random numbers and not in any way pre-specifying the driven line. As we have suggested in chapter 2 an alternative approach is that having established a base-line solution from random initial controls values, future inital values can be generated from small perturbations to the base-line solutions.

We have shown that the whilst changes in the path curvature of the driven line due to vehicle set-up can be significant, these changes are very small when seen in the plane of the track surface. We would expect then, that having established a base-line driven line, if the vehicle parameters do not change too much, only small perturbations to this line need be considered. We could therefore modify the displacement constraints to restrict the driven path to some region around the reference trajectory.

9.2.2.3 Alternative Approaches

Use of a general purpose NLP code to solve the problem in effect treats the objective and constraint functions as black boxes. The method applies no application specific knowledge in its search for a solution. It would be useful to consider how such knowledge could be used to help the optimizer reach solutions more quickly.

We have considered two bases for the independent variables (time/distance and frequency). It may be useful to consider if a more application specific basis can be found to represent the controls. For example we may consider that the braking torque for a particular corner should be a uni-modal function and therefore produce a basis that only allows the exploration of uni-modal functions. Alternatively, we may specify the independent variables as perturbations from some reference control values.

We have seen that tyre efficiency diagrams can be used to indicate that a trajectory is close to the limit performance of the vehicle. It would be interesting to apply constraints to the MTM algorithm that the tyres must achieve some minimum efficiency during key tyre limited parts of the manoeuvre.

In a similar fashion, constraints could be applied on the vehicle accelerations such that they are always close to the performance limit as indicated by a steady state g-g-v surface.

9.2.3 Finite Horizon

The Finite Horizon method was used to extend the MTM method for the time optimal solution of manoeuvres of arbitrary length. We have demonstated for a number of manoeuvres, that there exists a preview distance beyond which no meaningful decrease in manoeuvre time is found. Hence the Finite Horizon procedure appears to be a suitable approach for

the time optimal control of the race car. This approach also provides insight into the nature of the problem of finding an optimal driven line and speed profile. We have shown how preview distance affects the approach to a single corner and the compromise arising when one corner is followed immediately by another.

The motivation for use of the Finite Horizon approach as opposed to the approach used by Casanova [8] is that it does not require all of the model state variables to be exposed to the optimizer. This, at least in theory, allows the method to be used with arbitrarily complex vehicle models. We have demonstrated its use with a sprung car and with thermodynamic tyres.

The main drawback with the finite horizon approach is its inefficiency. The overlap between successive steps of the algorithm results in each control value being optimized several times before its final optimal value is chosen. This is highlighted in an exagertaed form in figure 6.7. This problem can be reduced if small compromises are made as follows:

- The algorithm can be assisted to find the driven line by placing 'road cones' on the race-track around a previously computed reference line

- At the first step of the finite horizon algorithm we must initialise all active vehicle control inputs with initial values. At subsequent steps we can choose to re-initialise all control values. Alternatively, for those control values that were active in the previous step we may keep the values from the previous steps optimal trajectory as initial values for the current step.

9.3 Comparison of QSS and Transient Lap time Simulation methods

A comparison of the results generated by the MTM method and a QSS type simulation method (Appendix D) has been performed for a simple 90° manoeuvre. The path curvature data for the QSS algorithm was provided from the optimal trajectories produced by the MTM method. In the author's experience, the results generated by QSS lap time simulation are very sensitive to the path curvature data used to describe the racing line. In the practical racing environment, path curvature is typically generated from on-car measurements of speed and lateral acceleration [8] and as such is representative only of the circumstances of the lap under which it was generated. We have demonstrated that changes in the optimal racing line due to vehicle roll moment distribution can cause a QSS manoeuvre time simulation to indicate inconsistent manoeuver time trends for changes in vehicle roll moment distribution. We have also shown that optimal vehicle state trajectories produced by the MTM method typically have significant yaw acceleration throughout a manoeuvre. The QSS method assumes that the yaw acceleration is zero at all times and this goes some way to explaining the differences between MTM and QSS results.

We discuss below the applications for lap time simulation codes and consider the applicability of MTM and QSS methods. Applications for a lap time simulation code can be split into two categories :

- A1) Vehicle design

- A2) Vehicle set-up at the trackside

Item A2 above can be further divided as follows:

- B1) The spring rates and tyre pressures must be chosen to keep the vehicle ride heights in ideal locations indicated by aerodynamic requirements.

- B2) The spring/damping rates and tyre pressures must be chosen to balance the trade-off between minimising ride height changes for aerodynamic requirements and minimising load fluctuations for tyre grip and time lost riding kerbs.

- B3) The spring/damping rates and tyre pressures must be chosen to produce desirable driveability characteristics to allow the driver access to limit performance.

- B4) The tyre pressures must be chosen to achieve optimal tyre performance

A cars basic capability envelope is dominated by its mass, engine torque curve and aerodynamics. This data can be readily represented by a g-g-v diagram (Appendix D). For this reason, QSS simulations are succesful at performing task B1 since the coarse effects of a ride height change are typically quite obvious in g-g-v data. The QSS approach is not directly applicable to item B2 since it cannot be used to simulate the car running over bumps / kerbs or rough surfaces.

A significant part of the process of setting up a car at the trackside is about making the car 'driveable' (item B3). That is, providing suitable handling qualities to allow the driver access to limit vehicle performance. We have shown through the use of vehicle stability derivatives (Appendix E) applied to time optimal trajectories that effect on vehicle stability of set-up parameters can be evaluated. The QSS method is not suitable for stability analysis since it cannot simulate transient vehicle behaviour.

In practice, in many categories of motor racing vehicle performance is dominated by tyre performance. In turn, tyre performance is dominated by tyre temperature. The major area opened up by a transient manoeuver time method for both vehicle design and set-up is that it allows the simulation of dynamic tyre temperature and pressure effects.

9.4 Driver Contribution to Vehicle performance

We have shown that manoeuvre time is reduced by:

- Using a controller bandwidth higher than that believed to be achievable by a human driver

- Allowing the vehicle to operate in open loop unstable modes

- Allowing the vehicle steering control moment to drop to zero

- Planning of the driven line and speed profile taking into account significant preview distances

The above findings go some way to explain the performance differences seen between professional drivers. The driver with the better ability to think / plan ahead, react quickly and tolerate a lower stability margin will be faster. We have given only slight consideration to the limits of the driver but any simulation capable of carrying out task B3 (above) should take into account the driver-vehicle system as one. A significant recommendation for further work is the development of metrics to indicate vehicle 'driveability'. One of the general challenges in F1 car set-up is the trade-off between entry stability and mid-corner understeer. In general, an increase in entry stability increases driver confidence and allows him/her to attack the corner. Set-up changes in this direction however, may result in an undesirable excess of stability in mid corner resulting in overall time lost through the corner. If a metric could be established indicating say minimum N_β and maximum N_δ during initial brake-and-turn-in, the car could be set-up using this metric as a constraint.

9.5 Sprung Vehicle Model

We have shown that the MTM method can operate succesfully using a sprung vehice model running over a planar track and also over a simulated kerb. The kerbing results give a good indication of the time lost in kerbing due to increasing values of heave and roll stiffness. At any given vehicle heave and roll stiffness there is a limit to the kerb height at which meaningful solutions can be found by the MTM method. This is one potential weakness of the NLP approach to the MTM problem. Riding high kerbs with a stiff vehicle set-up results in especially non-linear vehicle behaviour. As previously discussed, we have typically tended to use random initial control values. We have used stability constraints on understeer angle, vehicle side slip and tyre slip ratios to allow the optimizer to move from any arbitrary point to an optimal point without entering especially non-linear regions. An area for future work is to improve the performance of the algorithm in the application of sprung vehicles running over rough surfaces. The following are suggested:

- Use a close to optimal reference trajectory for the initial control values.
- Investigate the use of additional constraints such as wheel vertical acceleration.
- Use an NLP code that is more tolerant of noisy objective and constraint functions, for example the Dynamic-Q method [1],[37].

9.6 Thermal Tyre Model

As we have previously discussed, the tyre is arguably the most important and least understood aspect of the car. We have shown that the MTM method is capable of operation with a thermodynamic tyre model. Introduction of the model does not appear to have a strong influence on the non-linearity of the objective or constraint functions and the method remains capable of finding consistent and accurate solutions. This opens up considerable scope for further tyre model development. Some proposed additions / modifications to the tyre model are given below:

- Develop a more complex carcass model using Finite-Element methods.

- Use the enhanced carcass model to predict heating in the tyre due to deflection in the carcass when rolling and generating forces.
- Use the enhanced carcass model to more accurately describe lateral and combined force generation.

9.7 Additional Recommendations for Further Work

We propose here some further suggestions for further work not discussed so far.

9.7.1 Why does the car go faster ?

The MTM method allows the change in manoeuvre time associated with design and set-up parameters to be evaluated. We have also demonstrated at least to a limited extent that the stability of the vehicle operating at limit performance can be analysed. In order to make full use of the results it is necessary to know precisely why and how a parameter change has impacted the manoeuvre time. Hence we require methods to analyse and compare optimal vehicle state trajectories. The development of such methods could be as challenging as the development of the MTM method itself. The problem is highlighted in the following quote [54] :

> One of the drawbacks to using a complex computer program is that you seldom learn the "why" it came up with a particular "what"...A human may not be able to comprehend the underlying logic well enough to ever know. The intermediate steps and sub-conclusions may be buried in a stack of print-out sheets and plots. Sometimes it is more effective to use a simpler, more comprehensible analysis.

It is arguable that simpler analysis methods such as the steady state g-g-v diagram and Milliken moment method [36] should be used alongside the MTM method to assist in the interpretation of results.

9.7.2 Simultaneous Manoeuvre and Set-up Optimization

We have considered in this work the computation of optimal state trajectories for vehicles whose design and set-up parameters remain fixed throughout the optimization procedure. The MTM method could be modified to add vehicle design and set-up parameters to the optimizer independent variables as follows:

- Add a single value for a vehicle parameter e.g. longitudinal centre of mass to the independent variables
- Add a value for a vehicle parameter to the vehicle control values at each waypoint.

The first approach allows a parameter to be computed at its optimal value taking into account say an entire lap. The second approach (although not necessarily implementable on a real car) effectively adds additional controls to the control vector at each wayline.

The algorithm currently uses steer angle and throttle/brake but could be easily modified to allow other 'controls' such as roll balance, brake balance, roll/heave stiffness and damping etc. This second approach allows the optimal values for a parameter to be computed at each point around the lap.

Appendices

Appendix A

Vehicle Models

This appendix describes two vehicle models. The first is a quasi steady state model. The second is a fully dynamic car model with sprung and unsprung masses. The two vehicle models share a simple driveline and aerodynamic model.

A.1 Vehicle Axis System

The axis system used by the vehicle models is shown in figures A.1 and A.2. Rotation about the z axis is termed yaw and is denoted θ_y. The yaw and its time derivative yaw rate are positive in a lefthand corner. The lateral velocity is positive to the drivers left handside. Steer angle is positive in a lefthand corner. Rotation about the x axis is termed roll and denoted θ_r. It is positive in a lefthand corner. Rotation about the y axis is termed pitch and denoted θ_p. It is positive when braking.

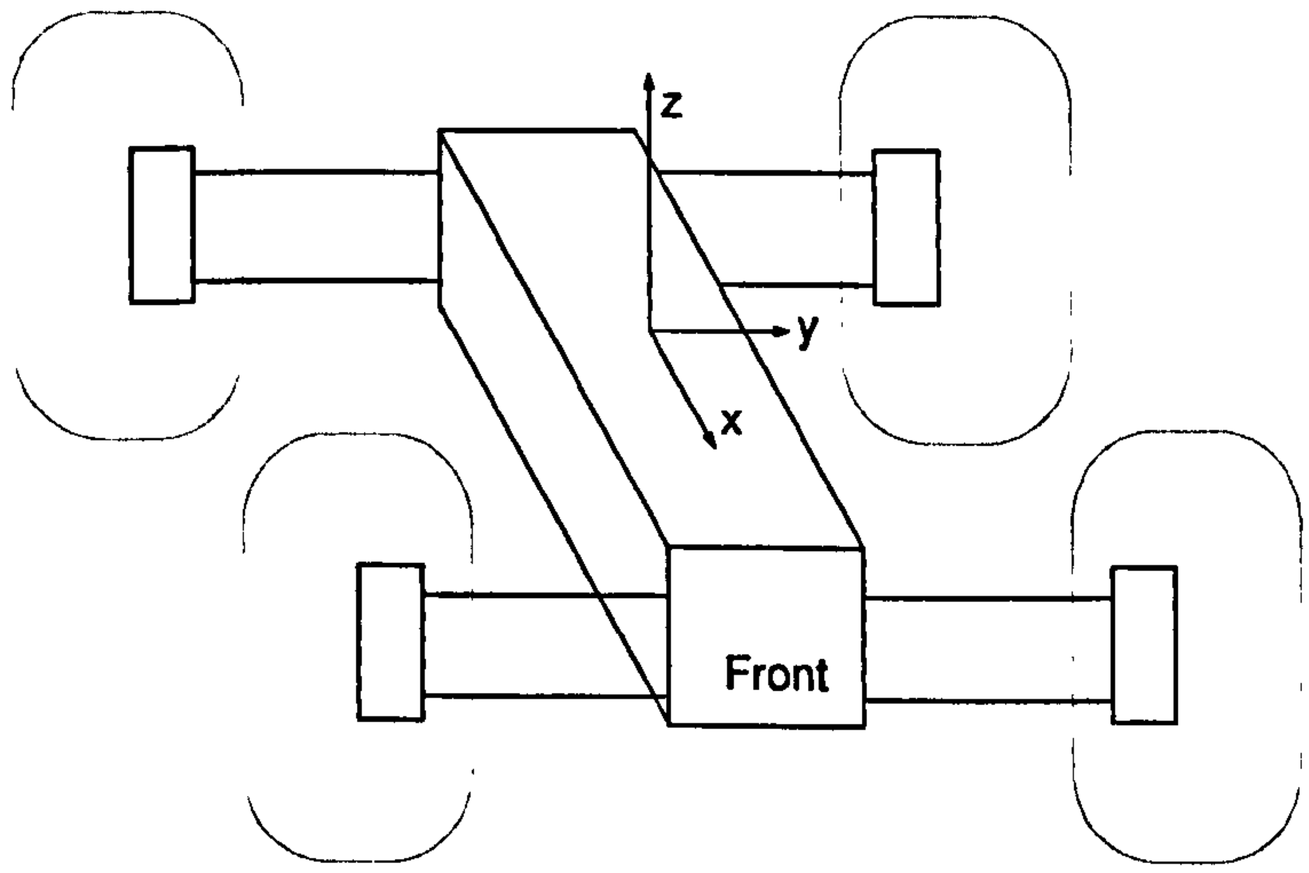


Figure A.1: Car Axes Sign Convention

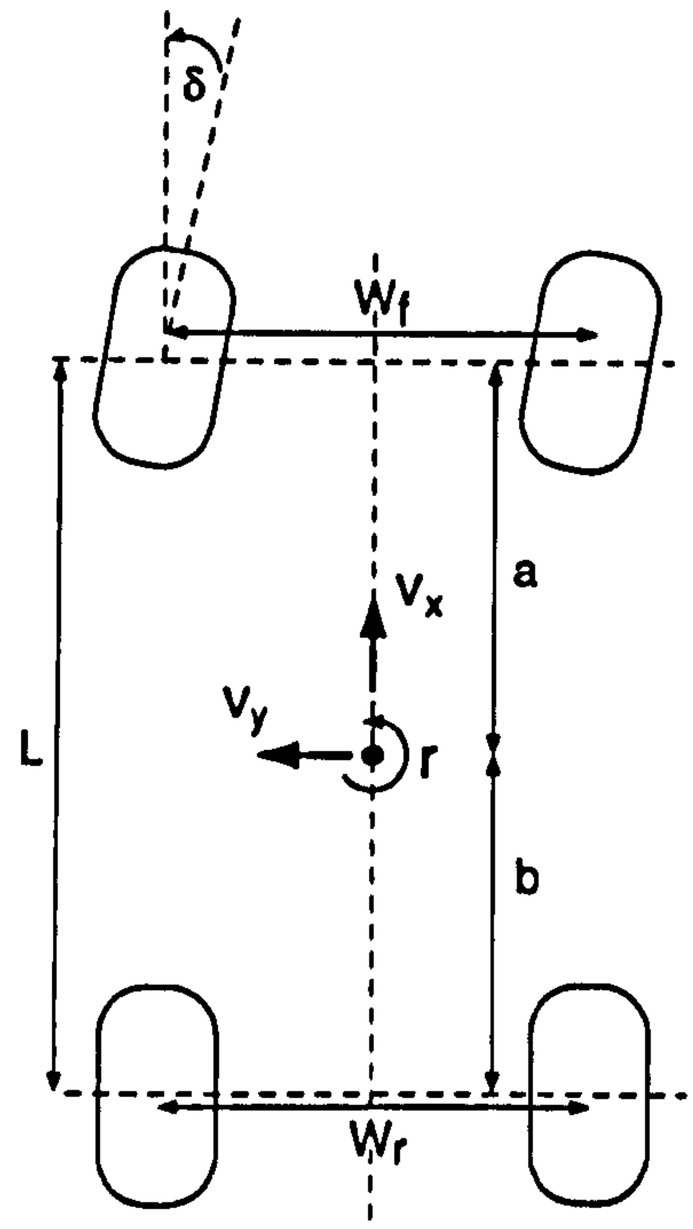


Figure A.2: Vehicle velocity axes Convention

The tyre slip quantities are computed as follows. Small angle assumptions are made.

Lateral velocity at centre of front and rear axle:

$$v_F = v_y + ar \tag{A.1}$$

$$v_R = v_y - br \tag{A.2}$$

Longitudinal velocity at centre of each wheel:

$$u_{LF} = v_x - \frac{rW_F}{2} \tag{A.3}$$

$$u_{RF} = v_x + \frac{rW_F}{2} \tag{A.4}$$

$$u_{LR} = v_x - \frac{rW_R}{2} \tag{A.5}$$

$$u_{RR} = v_x + \frac{rW_R}{2} \tag{A.6}$$

Slip angles:

$$\alpha_{LF} = \frac{v_F}{u_{LF}} - \delta \tag{A.7}$$

$$\alpha_{RF} = \frac{v_F}{u_{RF}} - \delta \tag{A.8}$$

$$\alpha_{LR} = \frac{v_R}{u_{LR}} \tag{A.9}$$

$$\alpha_{RR} = \frac{v_R}{u_{RR}} \tag{A.10}$$

Slip ratios:

$$\kappa_{LF} = \frac{\omega_{LF}}{u_{LF} R_F} - 1.0 \tag{A.11}$$

$$\kappa_{RF} = \frac{\omega_{RF}}{u_{RF} R_F} - 1.0 \tag{A.12}$$

$$\kappa_{LR} = \frac{\omega_{LR}}{u_{LR} R_R} - 1.0 \tag{A.13}$$

$$\kappa_{RR} = \frac{\omega_{RR}}{u_{RR} R_R} - 1.0 \tag{A.14}$$

A.2 Common Model Elements

The following elements are common to all models.

A.2.1 Aero Dynamics

The aerodynamic model is very simple. It is based solely on vehicle longitudinal speed as follows:

$$F_{df} = F_{dfref} \left(\frac{v_x}{u_{ref}} \right)^2 \tag{A.15}$$

$$F_{drag} = F_{dragref} \left(\frac{v_x}{u_{ref}} \right)^2 \tag{A.16}$$

F_{dfref} and $F_{dragref}$ are reference drag and downforce levels at the reference speed u_{ref}. The aerodynamic downforce is distributed between the two axles according to the aero balance D_{aero}.

$$F_{df_F} = D_{aero} F_{df} \tag{A.17}$$

$$F_{df_R} = (1.0 - D_{aero}) F_{df} \tag{A.18}$$

In both the QSS and sprung models the reaction points on the chassis of the front and rear aerodynamic load forces are at the centre of the axles. The load forces act in the earth reference frame negative z direction. The QSS model is effectively planar so the z heights of the aerodynamic force reaction points are not relevant. In the case of the sprung car they are applied at the chassis z zero height. Drag is applied directly to the centre of gravity of the chassis in the negative x direction in the earth reference xy plane.

A.2.2 Engine

The engine model is a simple two dimensional look up table indexed by engine revs and throttle position. Linear interpolation is used. The engine torque curves for maximum and minimum throttle positions are shown in figure A.3.

The rear wheels are driven via the gearbox and differential. Hence the engine speed is computed as:

$$\omega_{eng} = \frac{\tau(\omega_{LR} + \omega_{RR})}{2} \tag{A.19}$$

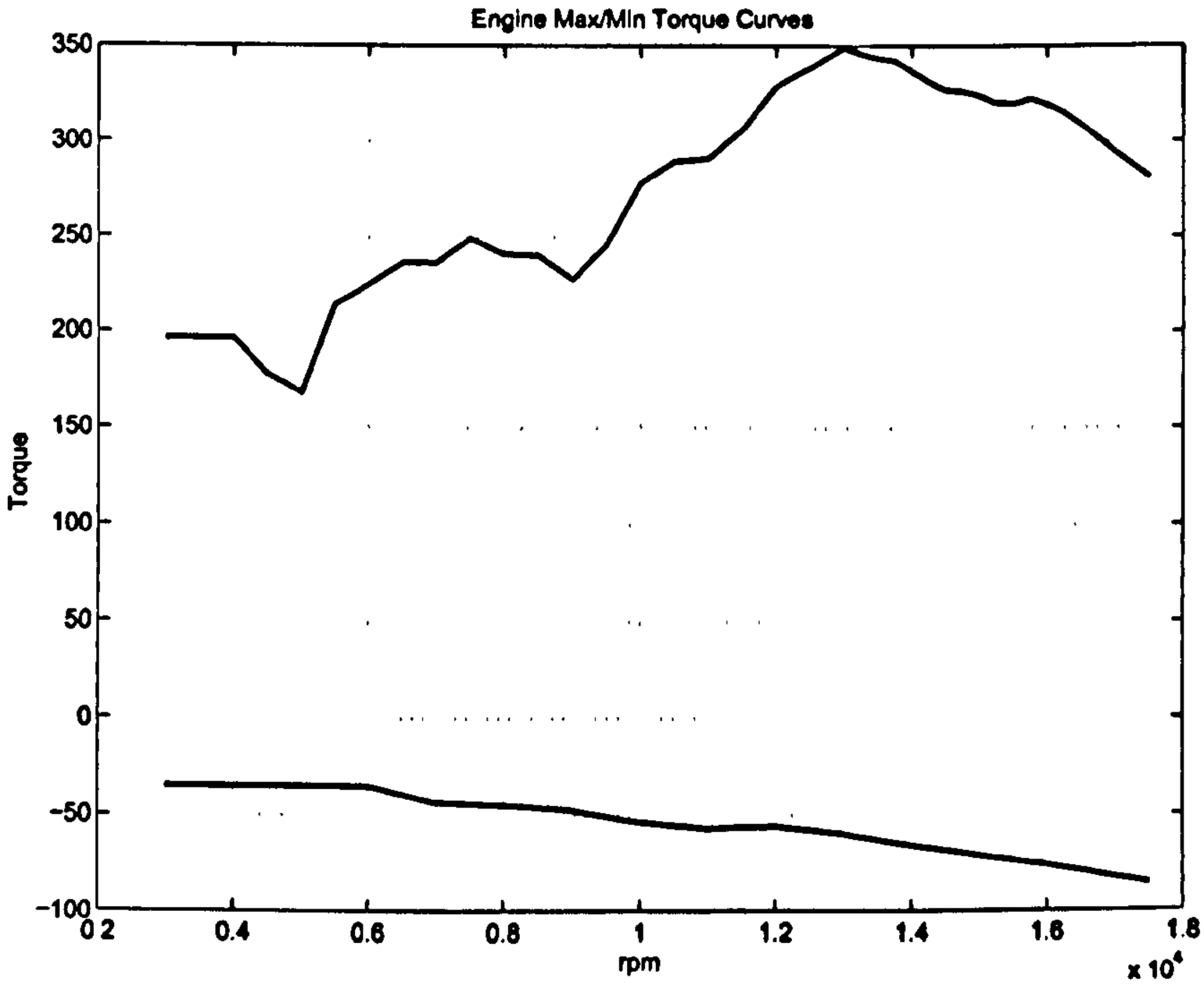


Figure A.3: Engine Max/Min Torque Curves

A.2.3 Brakes

Braking torques are applied to the wheels when the driver pedal input is negative. The total maximum available brake torque is given by $T_{brake_{MAX}}$

$$\begin{array}{ll} if(P < 0.0) & T_{brake} = -PT_{brake_{MAX}} \\ else & T_{brake} = 0.0 \end{array} \tag{A.20}$$

The brake torque is distributed between front and rear axles according to the brake balance D_{brake}.

$$T_{brake_F} = D_{brake} T_{brake} \tag{A.21}$$

$$T_{brake_R} = (1.0 - D_{brake}) T_{brake} \tag{A.22}$$

On each axle, the brake torque is distributed equally between the left and right wheels.

A.2.4 Gearbox

The gearbox is modelled as a simple torque/speed ratio. The gear ratio is automatically selected on the basis of vehicle longitudinal speed v_x. In the transition between gears, a CVT is emulated. This gearbox implementation is due to [8]. The clutch is not modelled.

Ratio	Speed
14.01900000	41.39788022
11.39050000	50.95095763
9.49210000	61.14104180
7.97330000	72.78753877
7.19400000	72.78753877
6.64440000	80.67234958

The gear ratio τ is computed as follows:

$$\begin{array}{l}\tau = \tau_1 + \\ \frac{(\tau_2 - \tau 1)}{2.0}(1.0 + \sin(\tan^{-1}(10.0(v_x - u_{\tau 0})))) + \\ \frac{(\tau_3 - \tau 2)}{2.0}(1.0 + \sin(\tan^{-1}(10.0(v_x - u_{\tau 1})))) + ...\end{array} \tag{A.23}$$

A.2.5 Differential

The differential is responsible for distributing the engine torque to the rear wheels. The model emulates a limited slip differential. It can be operated in one of three modes: Open, Limited Slip and Locked. The kinematic constraint imposed by the differential is as follows:

$$\frac{\omega_{eng}}{\tau} = \omega_{diff} = \frac{\omega_{LR} + \omega_{RR}}{2} \tag{A.24}$$

The torque at the input to the differential is as follows:

$$T_{diff_{in}} = T_{eng}\tau \tag{A.25}$$

The speed across the differential is defined as follows:

$$\Delta\omega_{diff} = \omega_{LR} - \omega_{RR} \tag{A.26}$$

A.2.5.1 Open

In the open case, the differential divides the engine torque equally between the left and right wheels.

$$T_{drive_{LR}} = T_{drive_{RR}} = \frac{T_{diff_{in}}}{2} \tag{A.27}$$

The torque transferred by the differential is zero:

$$\Delta T_{diff} = 0.0 \tag{A.28}$$

A.2.5.2 Limited Slip

In the limited slip mode, the model emulates the operation of a clutch between the left and right driveshatfs. The clutch transfers torque from the faster to the slower wheel. The torque transferred by the clutch is modelled as follows:

if $T_{diff_{in}} > 0$ then:

$$\Delta T_{diff} = (K_{diff_{PL}} - K_{diff_D} T_{diff_{in}})(sin(tan^{-1}(\Delta\omega_{diff}))(1 - cos(tan^{-1}(2\Delta\omega_{diff}))) \tag{A.29}$$

if $T_{diff_{in}} \leq 0$ then:

$$\Delta T_{diff} = (K_{diff_{PL}} + K_{diff_O} T_{diff_{in}})(sin(tan^{-1}(\Delta\omega_{diff}))(1 - cos(tan^{-1}(2\Delta\omega_{diff}))) \tag{A.30}$$

Parameter	Set-up A	Set-up B
$K_{diff_{PL}}$	20	20
K_{diff_D}	0.3	0.3
K_{diff_O}	0.3	0.7

Table A.1: Limited Slip Differential Set-up Parameters

A.2.5.3 Locked

In the locked case, the dynamic equations for the driveline are formulated with both left and right wheels combined into a single body.

A.2.6 Driveline and wheel Equations of motion

Rear wheel equations for open and limited slip differential:

$$\dot{\omega}_{LR} = \frac{T_{brake_{LR}} - R_R F_{x_{LR}} - \Delta T_{diff} + \frac{T_{diff_{in}}}{2}}{J_R + \frac{J_d \tau^2}{2}} \tag{A.31}$$

$$\dot{\omega}_{RR} = \frac{T_{brake_{RR}} - R_R F_{x_{RR}} + \Delta T_{diff} + \frac{T_{diff_{in}}}{2}}{J_R + \frac{J_d \tau^2}{2}} \tag{A.32}$$

Where J_R is the inertia of the rear wheel/tyre and J_d is the inertia of the remaining driveline/engine referred to the engine side of the gearbox. The torque at the wheels due to the brakes is given by $T_{brake_{LR}}$ and $T_{brake_{RR}}$. The rear wheel radius R_R assumes a constant value. The Rear axle equations for a locked differential are as follows:

$$\dot{\omega}_{LR} = \dot{\omega}_{RR} = \frac{T_{brake_R} - R_R F_{x_{LR}} - R_R F_{x_{RR}} + T_{diff_{in}}}{2J_R + J_d \tau^2} \tag{A.33}$$

The front wheel equations of motion are as follows:

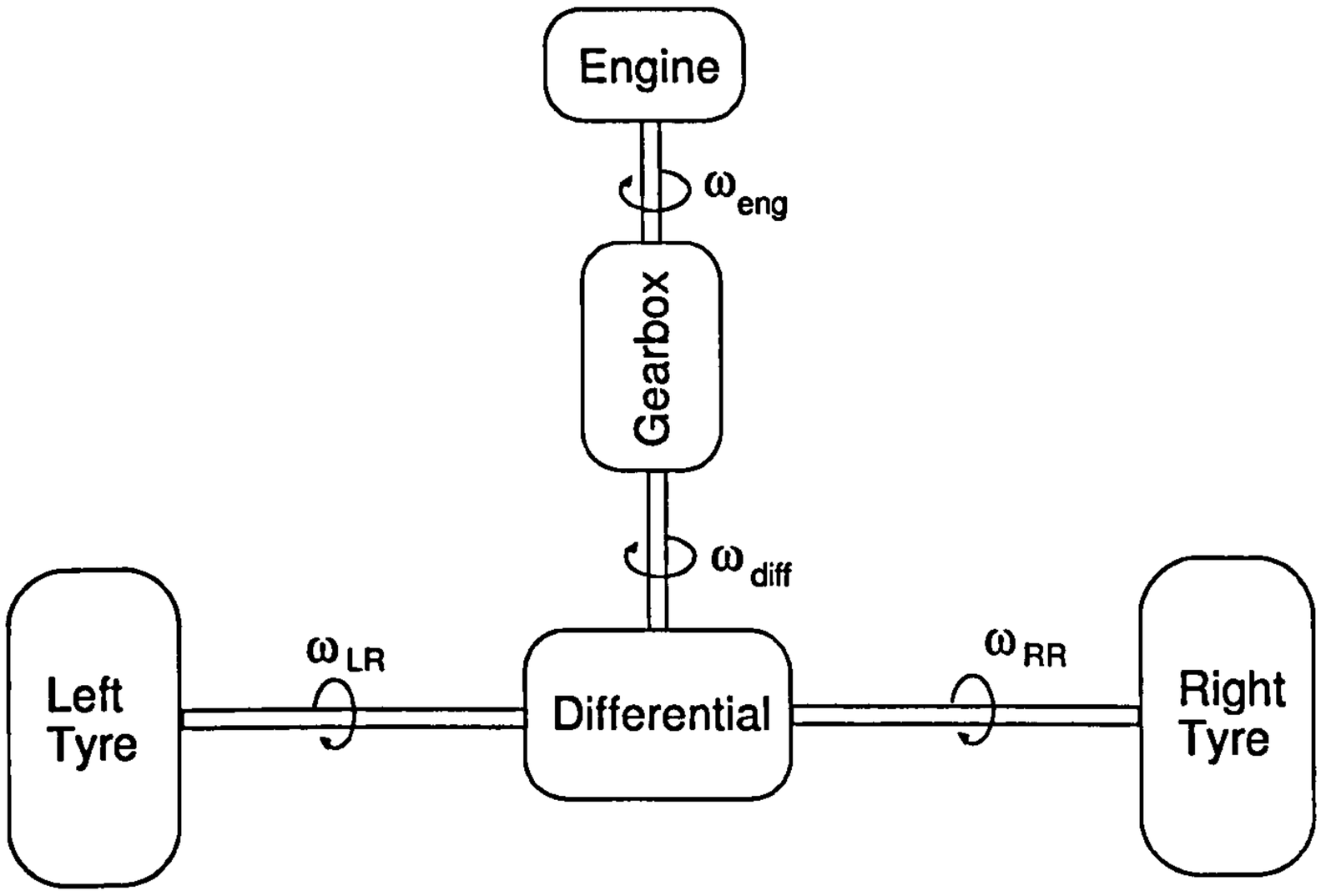


Figure A.4: Driveline Schematic

$$\dot{\omega}_{LF} = \frac{T_{brake_{LF}} - R_F F_{x_{LF}}}{J_F} \tag{A.34}$$

$$\dot{\omega}_{RF} = \frac{T_{brake_{RF}} - R_F F_{x_{RF}}}{J_F} \tag{A.35}$$

A.2.7 Tyre Forces

Two tyre models are used as described in appendix B and chapter 8. The longitudinal and lateral forces generated by each tyre are computed as a function of the following:

- Tyre vertical load
- Tyre slip angle
- Tyre sip ratio
- Tyre ground velocities in tyre x,y plane (Brush Model Only)

$$F_{x_{LF}} = TyreLongForce(F_{z_{LF}}, \alpha_{LF}, \kappa_{LF}) \tag{A.36}$$

$$F_{x_{RF}} = TyreLongForce(F_{z_{RF}}, \alpha_{RF}, \kappa_{RF}) \tag{A.37}$$

$$F_{x_{LR}} = TyreLongForce(F_{z_{LR}}, \alpha_{LR}, \kappa_{LR}) \tag{A.38}$$

$$F_{x_{RR}} = TyreLongForce(F_{z_{RR}}, \alpha_{RR}, \kappa_{RR}) \tag{A.39}$$

$$F_{y_{LF}} = TyreLatForce(F_{z_{LF}}, \alpha_{LF}, \kappa_{LF}) \tag{A.40}$$

$$F_{y_{RF}} = TyreLatForce(F_{z_{RF}}, \alpha_{RF}, \kappa_{RF}) \tag{A.41}$$

$$F_{y_{LR}} = TyreLatForce(F_{z_{LR}}, \alpha_{LR}, \kappa_{LR}) \tag{A.42}$$

$$F_{y_{RR}} = TyreLatForce(F_{z_{RR}}, \alpha_{RR}, \kappa_{RR}) \tag{A.43}$$

A.2.8 Steering

The front tyre lateral and longitudinal forces are resolved into the chassis axis by rotation by the steer angle δ.

$$F_{y_{LF_s}} = cos(\delta)F_{y_{LF}} + sin(\delta)F_{x_{LF}} \tag{A.44}$$

$$F_{x_{LF_s}} = cos(\delta)F_{x_{LF}} - sin(\delta)F_{y_{LF}} \tag{A.45}$$

$$F_{y_{RF_s}} = cos(\delta)F_{y_{RF}} + sin(\delta)F_{x_{RF}} \tag{A.46}$$

$$F_{x_{RF_s}} = cos(\delta)F_{x_{RF}} - sin(\delta)F_{y_{RF}} \tag{A.47}$$

A.3 Quasi Steady State Model

This model is similar to that formulated by Casanova [8]. The model consists of five bodies: a chassis and four wheels. The chassis is free to translate and rotate in the plane having two translational and one rotational degree of freedom. The wheels have a single rotational degree of freedom relative to the chassis. Gyroscopic forces from the roating wheels are not modelled.

The centre of gravity and roll centre heights are fixed i.e. do not vary with load or speed. The wheel loads are computed using a steady state approximation of lateral and longitudinal load transfer.

A.3.1 Lateral Load Transfer

Lateral acceleration is approximated as $A_y = v_x r$. From this, the total resultant lateral tyre force acting on the vehicle is estimated as $F_y = mA_y$. The tyre forces act at ground level resulting in a moment on the car about the x axis with a magnitude of $h_{cog}F_y$. This moment is balanced by lateral 'load' transfer. The total lateral load transfer is then computed using a simple steady state approximation as:

$$\Delta_Y = \frac{h_{cog}F_y}{0.5(W_F + W_R)} \tag{A.48}$$

Where W_F and W_R are the vehicle front and rear track. This is then distributed between the two axles based on the 'roll moment distribution' D_{roll} as follows:

$$\Delta_{Y_F} = D_{roll}\Delta_Y \tag{A.49}$$

$$\Delta_{Y_R} = (1.0 - D_{roll})\Delta_Y \tag{A.50}$$

As lateral acceleration increases, the load transfer on one axle may reach the point where one wheel becomes fully unloaded. Under these circumstances, any remaining load transfer must be taken up by the other axle.

A.3.2 Longitudinal Load Transfer

The longitudinal tyre forces are approximated as the braking torques and engine torque multiplied by the wheel radii to estimate a single resultant force F_x acting at ground level. The longitudinal load transfer is then calculated in a similar manner to the lateral case as shown in equation A.51.

$$\Delta_x = \frac{h_{cog} F_x}{L} \tag{A.51}$$

A.3.3 Wheel Loads

The wheel loads are computed from the following factors:

- The mass of the vehicle
- The aerodynamic load
- The lateral load transfer
- The longitudinal load transfer

We compute two sets of wheel loads. The first set are referred to as 'virtual' wheel loads. The virtual wheel loads become negative when a wheel is no longer in contact with the ground. In the case of the QSS model, this typically happens at high values of lateral load transfer. The virtual wheel loads are used in the computation of the wheel load constraints. The second set are referred to as the 'real' wheel loads. A real wheel load is equal to its corresponding virtual wheel load when the virtual wheel load is greater than or equal to zero and zero when the corresponding virtual wheel load is less than zero. The real wheel loads are used in the computation of tyre forces.

$$F_{zv_{LF}} = \frac{\frac{b}{L} Mg - \Delta_x + D_{aero} F_{df} - \Delta_{y_F}}{2} \tag{A.52}$$

$$F_{zv_{RF}} = \frac{\frac{b}{L} Mg - \Delta_x + D_{aero} F_{df} + \Delta_{y_F}}{2} \tag{A.53}$$

$$F_{zv_{LR}} = \frac{\frac{a}{L} Mg + \Delta_x + (1 - D_{aero}) F_{df} - \Delta_{y_R}}{2} \tag{A.54}$$

$$F_{zv_{RR}} = \frac{\frac{a}{L} Mg + \Delta_x + (1 - D_{aero}) F_{df} + \Delta_{y_R}}{2} \tag{A.55}$$

$$F_{z_{LF}} = F_{zv_{LF}} \quad when \quad F_{zv_{LF}} \geq 0 \tag{A.56}$$
$$F_{z_{LF}} = 0 \quad when \quad F_{zv_{LF}} < 0 \tag{A.57}$$
$$F_{z_{RF}} = F_{zv_{RF}} \quad when \quad F_{zv_{RF}} \geq 0 \tag{A.58}$$
$$F_{z_{RF}} = 0 \quad when \quad F_{zv_{RF}} < 0 \tag{A.59}$$
$$F_{z_{LR}} = F_{zv_{LR}} \quad when \quad F_{zv_{LR}} \geq 0 \tag{A.60}$$
$$F_{z_{LR}} = 0 \quad when \quad F_{zv_{LR}} < 0 \tag{A.61}$$
$$F_{z_{RR}} = F_{zv_{RR}} \quad when \quad F_{zv_{RR}} \geq 0 \tag{A.62}$$
$$F_{z_{RR}} = 0 \quad when \quad F_{zv_{RR}} < 0 \tag{A.63}$$
$$\tag{A.64}$$

A.3.4 Chassis equations of motion

Longitudinal acceleration:

$$\dot{v_x} = \frac{F_{xLF} + F_{xRF} + F_{xLR} + F_{xRR} - F_{drag}}{m} \tag{A.65}$$

Acceleration in lateral velocity:

$$\dot{v_y} = \frac{F_{yLF} + F_{yRF} + F_{yLR} + F_{yRR}}{m} \tag{A.66}$$

Yaw acceleration:

$$\dot{r} = \frac{b(F_{yLR} + F_{yRR}) - a(F_{yLF_s} + F_{yRF_s}) + 0.5W_R(F_{xRR} - F_{xLR}) + 0.5W_F(F_{xRF_s} - F_{xLF_s})}{I_{zz}} \tag{A.67}$$

A.4 Sprung Model

This section describes a sprung vehicle model developed using the multibody code Autosim. The model is described in chapter 7. A few additional notes and the Autosim source listing are provided here.

A.4.1 Sprung Model Tyre Forces

The vertical loads on the tyre are taken from the forces in the tyre vertical spring. Slip angles and ratios are computed as for the QSS model using the yaw rate, lateral and longitudinal velocities of the chassis body. Hence lateral velocity effects at the hub due to suspension velocity are neglected. As previously stated, camber angles are neglected so any dynamic camber effects are not considered.

A.4.2 Autosim Listing

```
;*****************************************************************************
;* Autosim Source file (LISP)
;*
;* Sprung car model with independent heave and roll dynamics
;* and double link suspension
;*
;* DPK 14th April 2005
;*****************************************************************************

(reset)
(si)
(add-gravity :direction -[nz])
(setsym *multibody-system-name* "F1 Car with separate heave and roll dynamics");
(setsym *target-language* C)
(setsym *Double-Precision* T)

;*****************************************************************************
;*       Define reference points of the system in its nominal configuration
;*****************************************************************************

(add-point N_AERO_FRONT :coordinates (tub_a 0.0 0.0)  :name "Front Aero action point")
(add-point N_AERO_REAR  :coordinates (-tub_b 0.0 0.0) :name "Rear Aero action point")

(add-point N_RR_PUP_TOP :coordinates (-tub_b  -0.05  0.2))
(add-point N_RR_PUP_BOT :coordinates (-tub_b  -0.05  0.0))
(add-point N_RR_UPR_TOP :coordinates (-tub_b  -0.725  0.2))
(add-point N_RR_UPR_HUB :coordinates (-tub_b  -0.725  0.1))
(add-point N_RR_UPR_BOT :coordinates (-tub_b  -0.725  0.0))
(add-point N_RR_CP      :coordinates (-tub_b  -0.725  rr_cp_height))

(add-point N_LR_PUP_TOP :coordinates (-tub_b   0.05  0.2))
(add-point N_LR_PUP_BOT :coordinates (-tub_b   0.05  0.0))
(add-point N_LR_UPR_TOP :coordinates (-tub_b   0.725  0.2))
(add-point N_LR_UPR_HUB :coordinates (-tub_b   0.725  0.1))
(add-point N_LR_UPR_BOT :coordinates (-tub_b   0.725  0.0))
(add-point N_LR_CP      :coordinates (-tub_b   0.725  lr_cp_height))

(add-point N_RF_PUP_TOP :coordinates ( tub_a  -0.05  0.2))
(add-point N_RF_PUP_BOT :coordinates ( tub_a  -0.05  0.0))
(add-point N_RF_UPR_TOP :coordinates ( tub_a  -0.75  0.2))
(add-point N_RF_UPR_HUB :coordinates ( tub_a  -0.75  0.1))
(add-point N_RF_UPR_BOT :coordinates ( tub_a  -0.75  0.0))
(add-point N_RF_CP      :coordinates ( tub_a  -0.75  rf_cp_height))

(add-point N_LF_PUP_TOP :coordinates ( tub_a   0.05  0.2))
(add-point N_LF_PUP_BOT :coordinates ( tub_a   0.05  0.0))
(add-point N_LF_UPR_TOP :coordinates ( tub_a   0.75  0.2))
(add-point N_LF_UPR_HUB :coordinates ( tub_a   0.75  0.1))
(add-point N_LF_UPR_BOT :coordinates ( tub_a   0.75  0.0))
(add-point N_LF_CP      :coordinates ( tub_a   0.75  lf_cp_height))

(add-point N_COFG    :coordinates ( 0      0  cg_height) :name "CG")
(add-point N_COFG_F  :coordinates ( tub_a  0  cg_height) :name "CG_F")
(add-point N_COFG_R  :coordinates (-tub_b  0  cg_height) :name "CG_R")

;*****************************************************************************
;                                   Define bodies
;*****************************************************************************
;* Chassis, uprights and suspension members

(add-body TF_FRAME
  :translate ([nx] [ny])
  :body-rotation-axes (z)
  :mass 0
  :parent N
  :name "TF_Frame")
```

```
(add-body TUB
  :translate z
  :body-rotation-axes (x y)
  :cm-coordinates N_COFG
  :parent TF_FRAME
  :name "Chassis")

(add-body RR_LWB :parent TUB  :mass 0   :inertia-matrix 0
                 :joint-coordinates N_RR_PUP_BOT  :body-rotation-axes x)
(add-body LR_LWB :parent TUB  :mass 0   :inertia-matrix 0
                 :joint-coordinates N_LR_PUP_BOT  :body-rotation-axes x)
(add-body RF_LWB :parent TUB  :mass 0   :inertia-matrix 0
                 :joint-coordinates N_RF_PUP_BOT  :body-rotation-axes x)
(add-body LF_LWB :parent TUB  :mass 0   :inertia-matrix 0
                 :joint-coordinates N_LF_PUP_BOT  :body-rotation-axes x)
(add-body RR_UWB :parent TUB  :mass 0   :inertia-matrix 0
                 :joint-coordinates N_RR_PUP_TOP  :body-rotation-axes x)
(add-body LR_UWB :parent TUB  :mass 0   :inertia-matrix 0
                 :joint-coordinates N_LR_PUP_TOP  :body-rotation-axes x)
(add-body RF_UWB :parent TUB  :mass 0   :inertia-matrix 0
                 :joint-coordinates N_RF_PUP_TOP  :body-rotation-axes x)
(add-body LF_UWB :parent TUB  :mass 0   :inertia-matrix 0
                 :joint-coordinates N_LF_PUP_TOP  :body-rotation-axes x)

(add-body LR_UPR :parent LR_LWB  :mass us_mass_rear    :inertia-matrix 0
                 :joint-coordinates N_LR_UPR_BOT  :body-rotation-axes x)
(add-body RR_UPR :parent RR_LWB  :mass us_mass_rear    :inertia-matrix 0
                 :joint-coordinates N_RR_UPR_BOT  :body-rotation-axes x)
(add-body LF_UPR :parent LF_LWB  :mass us_mass_front   :inertia-matrix 0
                 :joint-coordinates N_LF_UPR_BOT  :body-rotation-axes x)
(add-body RF_UPR :parent RF_LWB  :mass us_mass_front   :inertia-matrix 0
                 :joint-coordinates N_RF_UPR_BOT  :body-rotation-axes x)

;*****************************************************************************
;*                  Define moving reference points in bodies
;*****************************************************************************

(add-point RR_HUB     :body RR_UPR :coordinates N_RR_UPR_HUB)
(add-point RR_CP      :body RR_UPR :coordinates N_RR_CP)
(add-point RR_UPR_TOP :body RR_UPR :coordinates N_RR_UPR_TOP)
(add-point RR_UPR_BOT :body RR_UPR :coordinates N_RR_UPR_BOT)
(add-point RR_PUP_TOP :body TUB    :coordinates N_RR_PUP_TOP)
(add-point RR_PUP_BOT :body TUB    :coordinates N_RR_PUP_BOT)
(add-point RR_UWB_OUT :body RR_UWB :coordinates N_RR_UPR_TOP)

(add-point LR_HUB     :body LR_UPR :coordinates N_LR_UPR_HUB)
(add-point LR_CP      :body LR_UPR :coordinates N_LR_CP)
(add-point LR_UPR_TOP :body LR_UPR :coordinates N_LR_UPR_TOP)
(add-point LR_UPR_BOT :body LR_UPR :coordinates N_LR_UPR_BOT)
(add-point LR_PUP_TOP :body TUB    :coordinates N_LR_PUP_TOP)
(add-point LR_PUP_BOT :body TUB    :coordinates N_LR_PUP_BOT)
(add-point LR_UWB_OUT :body LR_UWB :coordinates N_LR_UPR_TOP)

(add-point RF_HUB     :body RF_UPR :coordinates N_RF_UPR_HUB)
(add-point RF_CP      :body RF_UPR :coordinates N_RF_CP)
(add-point RF_UPR_TOP :body RF_UPR :coordinates N_RF_UPR_TOP)
(add-point RF_UPR_BOT :body RF_UPR :coordinates N_RF_UPR_BOT)
(add-point RF_PUP_TOP :body TUB    :coordinates N_RF_PUP_TOP)
(add-point RF_PUP_BOT :body TUB    :coordinates N_RF_PUP_BOT)
(add-point RF_UWB_OUT :body RF_UWB :coordinates N_RF_UPR_TOP)

(add-point LF_HUB     :body LF_UPR :coordinates N_LF_UPR_HUB)
(add-point LF_CP      :body LF_UPR :coordinates N_LF_CP)
(add-point LF_UPR_TOP :body LF_UPR :coordinates N_LF_UPR_TOP)
(add-point LF_UPR_BOT :body LF_UPR :coordinates N_LF_UPR_BOT)
(add-point LF_PUP_TOP :body TUB    :coordinates N_LF_PUP_TOP)
(add-point LF_PUP_BOT :body TUB    :coordinates N_LF_PUP_BOT)
(add-point LF_UWB_OUT :body LF_UWB :coordinates N_LF_UPR_TOP)
```

```
(add-point AERO_FRONT :body TUB :coordinates N_AERO_FRONT)
(add-point AERO_REAR  :body TUB :coordinates N_AERO_REAR)

(add-point COFG       :body TUB :coordinates N_COFG)
(add-point COFG_F     :body TUB :coordinates N_COFG_F)
(add-point COFG_R     :body TUB :coordinates N_COFG_R)

;*****************************************************************************
;* Apply constraint between top wishbone outboard and top of upright
;*****************************************************************************
(no-movement RR_UWB_OUT RR_UPR_TOP [nz])
(no-movement RR_UWB_OUT RR_UPR_TOP [ny])
(no-movement LR_UWB_OUT LR_UPR_TOP [nz])
(no-movement LR_UWB_OUT LR_UPR_TOP [ny])
(no-movement RF_UWB_OUT RF_UPR_TOP [nz])
(no-movement RF_UWB_OUT RF_UPR_TOP [ny])
(no-movement LF_UWB_OUT LF_UPR_TOP [nz])
(no-movement LF_UWB_OUT LF_UPR_TOP [ny])

;*****************************************************************************
;*                              Define forces
;*****************************************************************************
;* Suspension forces

;(add-strut LF_SUSP_FORCE :magnitude "lf_susp_force" :point1 LF_PUP_TOP :point2 LF_UPR_BOT)
;(add-strut RF_SUSP_FORCE :magnitude "rf_susp_force" :point1 RF_PUP_TOP :point2 RF_UPR_BOT)
;(add-strut LR_SUSP_FORCE :magnitude "lr_susp_force" :point1 LR_PUP_TOP :point2 LR_UPR_BOT)
;(add-strut RR_SUSP_FORCE :magnitude "rr_susp_force" :point1 RR_PUP_TOP :point2 RR_UPR_BOT)


(add-moment LF_SUSP_FORCE :name "LF suspension torque" :direction [TUBx] :magnitude "lf_susp_force" :body1 TUB :body2 LF_LWB)
(add-moment RF_SUSP_FORCE :name "RF suspension torque" :direction [TUBx] :magnitude "rf_susp_force" :body1 TUB :body2 RF_LWB)
(add-moment LR_SUSP_FORCE :name "LR suspension torque" :direction [TUBx] :magnitude "lr_susp_force" :body1 TUB :body2 LR_LWB)
(add-moment RR_SUSP_FORCE :name "RR suspension torque" :direction [TUBx] :magnitude "rr_susp_force" :body1 TUB :body2 RR_LWB)


;*  Tyre Fz force

(add-line-force F_RR_Fz :direction [TF_FRAMEz] :magnitude "force_rr_Fz" :point1 RR_CP)
(add-line-force F_LR_Fz :direction [TF_FRAMEz] :magnitude "force_lr_Fz" :point1 LR_CP)
(add-line-force F_RF_Fz :direction [TF_FRAMEz] :magnitude "force_rf_Fz" :point1 RF_CP)
(add-line-force F_LF_Fz :direction [TF_FRAMEz] :magnitude "force_lf_Fz" :point1 LF_CP)

;* Tyre Fx force

(add-line-force F_RR_Fx :direction [TF_FRAMEx] :magnitude "tyre_rr_Fx" :point1 RR_CP)
(add-line-force F_LR_Fx :direction [TF_FRAMEx] :magnitude "tyre_lr_Fx" :point1 LR_CP)
(add-line-force F_RF_Fx :direction [TF_FRAMEx] :magnitude "tyre_rf_Fx" :point1 RF_CP)
(add-line-force F_LF_Fx :direction [TF_FRAMEx] :magnitude "tyre_lf_Fx" :point1 LF_CP)

;* Tyre Fy force

(add-line-force F_RR_Fy :direction [TF_FRAMEy] :magnitude "tyre_rr_Fy" :point1 RR_CP)
(add-line-force F_LR_Fy :direction [TF_FRAMEy] :magnitude "tyre_lr_Fy" :point1 LR_CP)
(add-line-force F_RF_Fy :direction [TF_FRAMEy] :magnitude "tyre_rf_Fy" :point1 RF_CP)
(add-line-force F_LF_Fy :direction [TF_FRAMEy] :magnitude "tyre_lf_Fy" :point1 LF_CP)

;* Aerodynamic forces

(add-line-force AERO_DRAG_F      :direction [TF_FRAMEx] :magnitude "aero_drag"        :point1 COFG)
(add-line-force AERO_DOWNFORCE_F :direction [TF_FRAMEz] :magnitude "aero_downforce_f" :point1 AERO_FRONT)
(add-line-force AERO_DOWNFORCE_R :direction [TF_FRAMEz] :magnitude "aero_downforce_r" :point1 AERO_REAR)

;*****************************************************************************
;*                            Define Dynamic variables
;*****************************************************************************
(add-variables dyvars real force_rr_Fz)
(add-variables dyvars real force_Lr_Fz)
```

```
(add-variables dyvars real force_rf_Fz)
(add-variables dyvars real force_Lf_Fz)
(add-variables dyvars real tyre_rr_Fx)
(add-variables dyvars real tyre_lr_Fx)
(add-variables dyvars real tyre_rf_Fx)
(add-variables dyvars real tyre_lf_Fx)
(add-variables dyvars real tyre_rr_Fy)
(add-variables dyvars real tyre_lr_Fy)
(add-variables dyvars real tyre_rf_Fy)
(add-variables dyvars real tyre_lf_Fy)
(add-variables dyvars real aero_drag)
(add-variables dyvars real aero_downforce_f)
(add-variables dyvars real aero_downforce_r)
(add-variables dyvars real rr_cp_height)
(add-variables dyvars real lr_cp_height)
(add-variables dyvars real rf_cp_height)
(add-variables dyvars real lf_cp_height)
(add-variables dyvars real rr_susp_force)
(add-variables dyvars real lr_susp_force)
(add-variables dyvars real rf_susp_force)
(add-variables dyvars real lf_susp_force)

;*****************************************************************************
;*    Define external subroutine to compute forces and tyre loaded radii
;*****************************************************************************

(add-subroutine difeqn compute_forces("t"
   &tyre_rr_Fx
   &tyre_lr_Fx
   &tyre_rf_Fx
   &tyre_lf_Fx
   &tyre_rr_Fy
   &tyre_lr_Fy
   &tyre_rf_Fy
   &tyre_lf_Fy
   &force_rr_Fz
   &force_lr_Fz
   &force_rf_Fz
   &force_lf_Fz
   &aero_drag
   &aero_downforce_f
   &aero_downforce_r
   &rr_susp_force
   &lr_susp_force
   &rf_susp_force
   &lf_susp_force
  "dot([nz],pos(RR_HUB))"
  "dot([nz],vel(RR_HUB))"
  "dot([nz],pos(LR_HUB))"
  "dot([nz],vel(LR_HUB))"
  "dot([nz],pos(RF_HUB))"
  "dot([nz],vel(RF_HUB))"
  "dot([nz],pos(LF_HUB))"
  "dot([nz],vel(LF_HUB))"
  "dot([nz],pos(COFG))"
  "dot([nz],pos(RR_PUP_BOT))"
  "dot([nz],pos(LR_PUP_BOT))"
  "dot([nz],pos(RF_PUP_BOT))"
  "dot([nz],pos(LF_PUP_BOT))"
  "dot([ny],pos(RR_CP))"))


;*****************************************************************************
;*                Define output points for animation and debugging
;*****************************************************************************

(add-out "dot([nx], pos(LF_HUB))" "N_LF_HUB_X")
(add-out "dot([ny], pos(LF_HUB))" "N_LF_HUB_Y")
(add-out "dot([nz], pos(LF_HUB))" "N_LF_HUB_Z")
```

```
(add-out "dot([nx], pos(RF_HUB))" "N_RF_HUB_X")
(add-out "dot([ny], pos(RF_HUB))" "N_RF_HUB_Y")
(add-out "dot([nz], pos(RF_HUB))" "N_RF_HUB_Z")
(add-out "dot([nx], pos(LR_HUB))" "N_LR_HUB_X")
(add-out "dot([ny], pos(LR_HUB))" "N_LR_HUB_Y")
(add-out "dot([nz], pos(LR_HUB))" "N_LR_HUB_Z")
(add-out "dot([nx], pos(RR_HUB))" "N_RR_HUB_X")
(add-out "dot([ny], pos(RR_HUB))" "N_RR_HUB_Y")
(add-out "dot([nz], pos(RR_HUB))" "N_RR_HUB_Z")

(add-out "dot([nx], pos(LF_CP))"  "N_LF_CP_X")
(add-out "dot([ny], pos(LF_CP))"  "N_LF_CP_Y")
(add-out "dot([nz], pos(LF_CP))"  "N_LF_CP_Z")
(add-out "dot([nx], pos(RF_CP))"  "N_RF_CP_X")
(add-out "dot([ny], pos(RF_CP))"  "N_RF_CP_Y")
(add-out "dot([nz], pos(RF_CP))"  "N_RF_CP_Z")
(add-out "dot([nx], pos(LR_CP))"  "N_LR_CP_X")
(add-out "dot([ny], pos(LR_CP))"  "N_LR_CP_Y")
(add-out "dot([nz], pos(LR_CP))"  "N_LR_CP_Z")
(add-out "dot([nx], pos(RR_CP))"  "N_RR_CP_X")
(add-out "dot([ny], pos(RR_CP))"  "N_RR_CP_Y")
(add-out "dot([nz], pos(RR_CP))"  "N_RR_CP_Z")

(add-out "dot([nx], pos(RR_PUP_TOP))" "N_RR_PUP_TOP_X")
(add-out "dot([ny], pos(RR_PUP_TOP))" "N_RR_PUP_TOP_Y")
(add-out "dot([nz], pos(RR_PUP_TOP))" "N_RR_PUP_TOP_Z")
(add-out "dot([nx], pos(LR_PUP_TOP))" "N_LR_PUP_TOP_X")
(add-out "dot([ny], pos(LR_PUP_TOP))" "N_LR_PUP_TOP_Y")
(add-out "dot([nz], pos(LR_PUP_TOP))" "N_LR_PUP_TOP_Z")
(add-out "dot([nx], pos(RF_PUP_TOP))" "N_RF_PUP_TOP_X")
(add-out "dot([ny], pos(RF_PUP_TOP))" "N_RF_PUP_TOP_Y")
(add-out "dot([nz], pos(RF_PUP_TOP))" "N_RF_PUP_TOP_Z")
(add-out "dot([nx], pos(LF_PUP_TOP))" "N_LF_PUP_TOP_X")
(add-out "dot([ny], pos(LF_PUP_TOP))" "N_LF_PUP_TOP_Y")
(add-out "dot([nz], pos(LF_PUP_TOP))" "N_LF_PUP_TOP_Z")

(add-out "dot([nx], pos(RR_UPR_TOP))" "N_RR_UPR_TOP_X")
(add-out "dot([ny], pos(RR_UPR_TOP))" "N_RR_UPR_TOP_Y")
(add-out "dot([nz], pos(RR_UPR_TOP))" "N_RR_UPR_TOP_Z")
(add-out "dot([nx], pos(LR_UPR_TOP))" "N_LR_UPR_TOP_X")
(add-out "dot([ny], pos(LR_UPR_TOP))" "N_LR_UPR_TOP_Y")
(add-out "dot([nz], pos(LR_UPR_TOP))" "N_LR_UPR_TOP_Z")
(add-out "dot([nx], pos(RF_UPR_TOP))" "N_RF_UPR_TOP_X")
(add-out "dot([ny], pos(RF_UPR_TOP))" "N_RF_UPR_TOP_Y")
(add-out "dot([nz], pos(RF_UPR_TOP))" "N_RF_UPR_TOP_Z")
(add-out "dot([nx], pos(LF_UPR_TOP))" "N_LF_UPR_TOP_X")
(add-out "dot([ny], pos(LF_UPR_TOP))" "N_LF_UPR_TOP_Y")
(add-out "dot([nz], pos(LF_UPR_TOP))" "N_LF_UPR_TOP_Z")

;*****************************************************************************
;*                        Create C code for the model
;*****************************************************************************

(finish t)
(write-to-file write-difeqn "c:\\lora\\f1d2\\f1car2_difeqn.c")
(write-to-file write-update "c:\\lora\\f1d2\\f1car2_update.c")
(write-to-file write-precmp "c:\\lora\\f1d2\\f1car2_precmp.c")
(write-to-file write-output "c:\\lora\\f1d2\\f1car2_out.c")
(write-to-file write-h      "c:\\lora\\f1d2\\f1car2.h")
(write-to-file write-init   "c:\\lora\\f1d2\\f1car2_init.c")
(write-to-file write-initnr "c:\\lora\\f1d2\\f1car2_initnr.c")
(write-to-file write-newtr1 "c:\\lora\\f1d2\\f1car2_newtr1.c")
(write-to-file write-sim    "c:\\lora\\f1d2\\f1car2_sim.c")
(format T "~%Writing equations file~%")
(write-to-file write-eqs "c:\\lora\\f1d2\\f1car.doc")
```

A.5 Vehicle Parameterization and Reference Configurations

Differential operating mode, open, locked or limited slip.
Diff parameters:

A.5.1 Common Parameters

Symbol	Name	Default Value
M	Total vehicle mass	620 kg
I_{zz}	Total vehicle moment of inertia about z axis	720
D_{aero}	Aero balance	0.4
D_{brake}	Brake balance	0.55
D_{mass}	Longitudinal mass distribution	0.475
$u_{ref.}$	Speed reference for aerodynamics	240 km/h

A.5.2 QSS Model

Symbol	Name	Default Value
D_{RC}	Roll moment distribution	
F_{dfref}	Reference aerodynamic downforce	9983.5 N
$F_{dragref}$	Reference aerodynamic drag	3833 N

A.5.2.1 QSS Setup A

Symbol	Name	Value
D_{RC}	Roll moment distribution	0.55

A.5.2.2 QSS Setup B

Symbol	Name	Value
D_{RC}	Roll moment distribution	0.65

A.5.3 Sprung Model

Symbol	Name	Default Value
M	Total vehicle mass	620 kg
M_s	Vehicle sprung mass	kg
M_{uF}	Vehicle unsprung mass (per wheel)at rear axle	kg
M_{uR}	Vehicle unsprung mass (per wheel)at front	kg
F_{dfref}	Reference aerodynamic downforce	12262.5 N
$F_{dragref}$	Reference aerodynamic drag	3335.4 N

Appendix B

Basic Tyre Model

Tyres in general and racing tyres in particular are the most complex aspect of the vehicle. The F1 racing tyre sees a wide range of loads and a wide range of temperatures. The following tyre model attempts to model the effect of load but temperature effects are considered in a more complex model developed in chapter 8.

The model is a 'Pacejka like' [30] model but with a considerably reduced number of parameters. The model allows the user to specify the value and location of the peak coefficient of friction (μ)in each of the tyre axes. This peak value and location are linearly scheduled with the tyre vertical load. The model also allows the specification of a simple 'shape factor' that determines the slip stiffness of the tyre at zero slip and the way the tyre 'rolls off' after peak slip has been exceeded. The combined slip condition is handled by a simple 'friction circle' procedure.

Given a tyre vertical load F_z and a slip ratio κ and slip angle α the tyre Lateral and Longitudinal forces are computed as follows:

Symbol	Description
F_{z1}	Reference Load 1 (N)
F_{z2}	Reference Load 2 (N)
μ_{xmax1}	Max Longitudinal μ at ref Load 1
μ_{xmax2}	Max Longitudinal μ at ref Load 2
κ_{max1}	Slip ratio for max Longitudinal μ at ref Load 1
κ_{max2}	Slip ratio for max Longitudinal μ at ref Load 2
Q_x	Longitudinal Shape Factor
μ_{ymax1}	Max Lateral μ at ref Load 1
μ_{ymax2}	Max Lateral μ at ref Load 2
α_{max1}	Slip angle for max Lateral μ at ref Load 1
α_{max2}	Slip angle for max Lateral μ at ref Load 2
Q_y	Lateral Shape Factor

Table B.1: Tyre Parameters

Work out the value and location of the lateral and longitudinal peaks μ_{xmax},κ_{max},μ_{ymax} and α_{max} using linear interpolation.

$$\mu_{xmax} = (F_z - F_{z1})\frac{(\mu_{xmax2} - \mu_{xmax1})}{(F_{z2} - F_{z1})} + \mu_{xmax1} \tag{B.1}$$

$$\mu_{ymax} = (F_z - F_{z1})\frac{(\mu_{ymax2} - \mu_{ymax1})}{(F_{z2} - F_{z1})} + \mu_{ymax1} \tag{B.2}$$

$$\kappa_{max} = (F_z - F_{z1})\frac{(\kappa_{max2} - \kappa_{max1})}{(F_{z2} - F_{z1})} + \kappa_{max1} \tag{B.3}$$

$$\alpha_{max} = (F_z - F_{z1})\frac{(\alpha_{max2} - \alpha_{max1})}{(F_{z2} - F_{z1})} + \alpha_{max1} \tag{B.4}$$

Normalise κ and α:

$$\kappa_{norm} = \frac{\kappa}{\kappa_{max}} \tag{B.5}$$

$$\alpha_{norm} = \frac{\alpha}{\alpha_{max}} \tag{B.6}$$

Friction circle rule:

$$\rho = \sqrt{(\alpha_{norm}^2 + \kappa_{norm}^2)} \tag{B.7}$$

Lateral and Longitudinal Scaling Factors:

$$S_x = \frac{\pi}{2tan^{-1}(Q_x)} \tag{B.8}$$

$$S_y = \frac{\pi}{2tan^{-1}(Q_y)} \tag{B.9}$$

$$\mu_x = \mu_{xmax} sin(Q_x tan^{-1}(S_x \rho)) \tag{B.10}$$

$$\mu_y = \mu_{ymax} sin(Q_y tan^{-1}(S_y \rho)) \tag{B.11}$$

Finally compute Longitudinal and Lateral forces:

$$F_x = \frac{F_z \kappa_n}{\rho} \tag{B.12}$$

$$F_y = \frac{F_z \alpha_n}{\rho} \tag{B.13}$$

The plot below shows an example μ / slip curve with load as a parameter. The plot shows how the μ peak occurs at a lower slip as the load increases. This is to intended to emulate the effect of the lengthening of the contact patch that occurs approximately in proportion

Symbol	Description
λ_{Fx}^{front}	Front tyre longitudinal force scale factor
λ_{Fy}^{front}	Front tyre lateral force scale factor
λ_{Fx}^{rear}	Rear tyre longitudinal force scale factor
λ_{Fy}^{rear}	Rear tyre lateral force scale factor

Table B.2: Test Configuration

to load. A longer contact patch results in more rubber in shear, hence more force for an equivalent slip ratio.

The plot below shows the impact of the 'shape factor'. The more rapidly the tyre rolls of around the peak μ value, the more difficult it becomes for the driver to balance the car at optimum acceleration / deceleration.

This model has the advantage of having relatively few parameters but allowing easy control over such parameters as peak mu and slip stiffness.

The model also allows easy computation of the slip ratio for peak mu. This can be helpful when analyzing the performance of a manoeuvre time optimizer.

B.0.3.1 Tyre Force Scale Factors

When studying the effect of tyre parameters on vehicle performance it is useful to be able to apply a simple scaling factor to the tyre forces. This allows for example, the study of the effect of a reduction in rear tyre grip on vehicle stability. To support this, the tyre model applies a separate scale factor to the lateral and longitudinal forces produced by the tyre.

The default value for the grip scaler is 1. In this case the grip scaler has no effect on the tyre force. To reduce the tyre force, a grip scaler with a value less than 1 is used e.g. a value of 0.98 will reduce the tyre force by 2%.

B.0.3.2 Tyre Parameterizations

Table B.3 shows the tyre parameters for the two tyre types used in this thesis.

Symbol	Set A Front	Set B Front	Set A Rear	Set B Rear
F_{z1}	2000 N	2000 N	2000N	2000N
F_{z2}	6000 N	6000 N	6000N	6000N
μ_{xmax1}	2.0	2.0	2.0	2.0
μ_{xmax2}	1.9	1.7	1.9	1.9
κ_{max1}	11%	13%	11%	12%
κ_{max2}	10%	11%	10%	10%
Q_x	1.9	1.5	1.9	1.5
μ_{ymax1}	1.9	1.8	1.9	1.7
μ_{ymax2}	1.8	1.5	1.8	1.4
α_{max1}	9 degress	5 degrees	8 degrees	3 degrees
α_{max2}	8 degrees	2 degrees	7 degrees	10 degrees
Q_y	1.9	1.5	1.9	1.9

Table B.3: Tyre Parameterisation

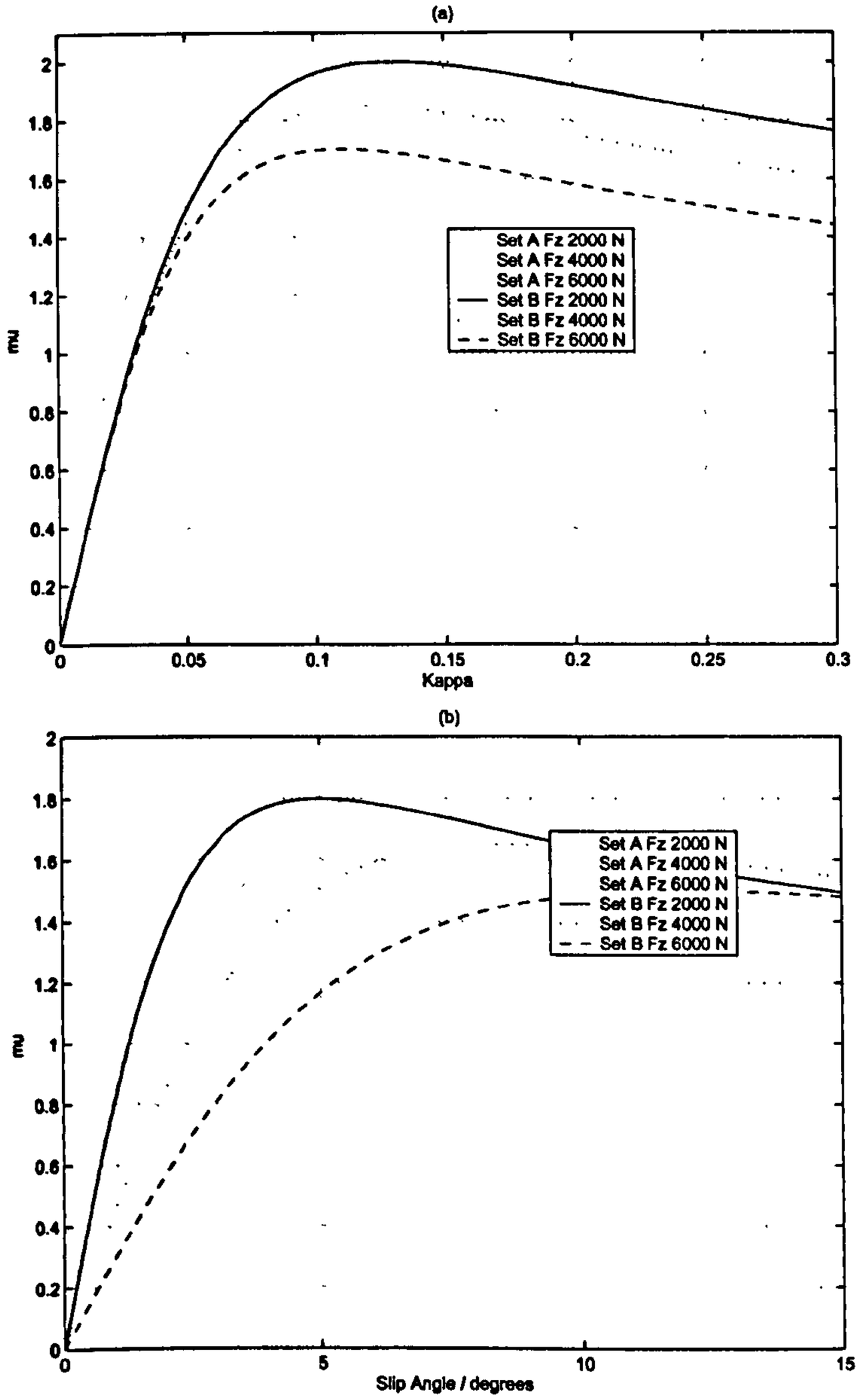


Figure B.1: Mu Slip Curves for Front Tyres (a) Longitudinal (b) Lateral

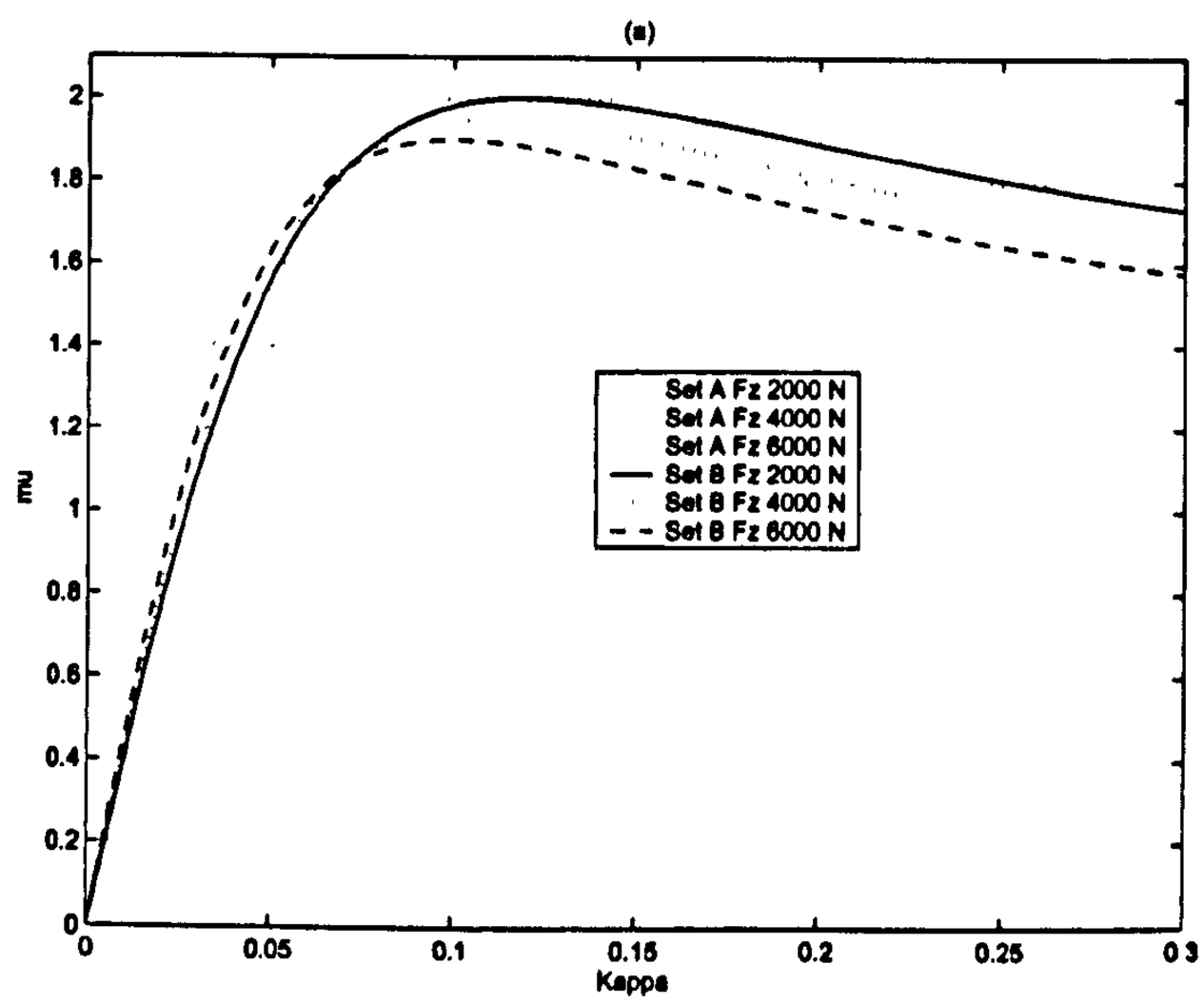


Figure B.2: Mu Slip Curves for Rear Tyres (a) Longitudinal (b) Lateral

B.0.3.3 Computing tyre efficiency

In order to help validate the solutions produced by the manoeuvre time optimizer and to understand the impact of vehicle set-up changes on tyre performance, it is helpful to produce plots of tyre utilisation. The basic tyre model described here lends itself particularly readily to this process. At a given load, it is possible to calculate the maximum pure lateral and longitudinal forces as shown in equations B.14 and B.15.

$$F_{xmax} = \mu_{xmax} F_z \tag{B.14}$$

$$F_{ymax} = \mu_{ymax} F_z \tag{B.15}$$

The actual tyre lateral and longitudinal forces can then be normalised by dividing by the corresponding maximum force. This effectively scales the tyre maximum force ellipse to a unit circle (see figure B.3). The length of the normalised tyre force vector is then a good indication of the useage of the full tyre potential. This number lies in the range $[0, 1]$ with 1 indicating that the full tyre potential has been exploited.

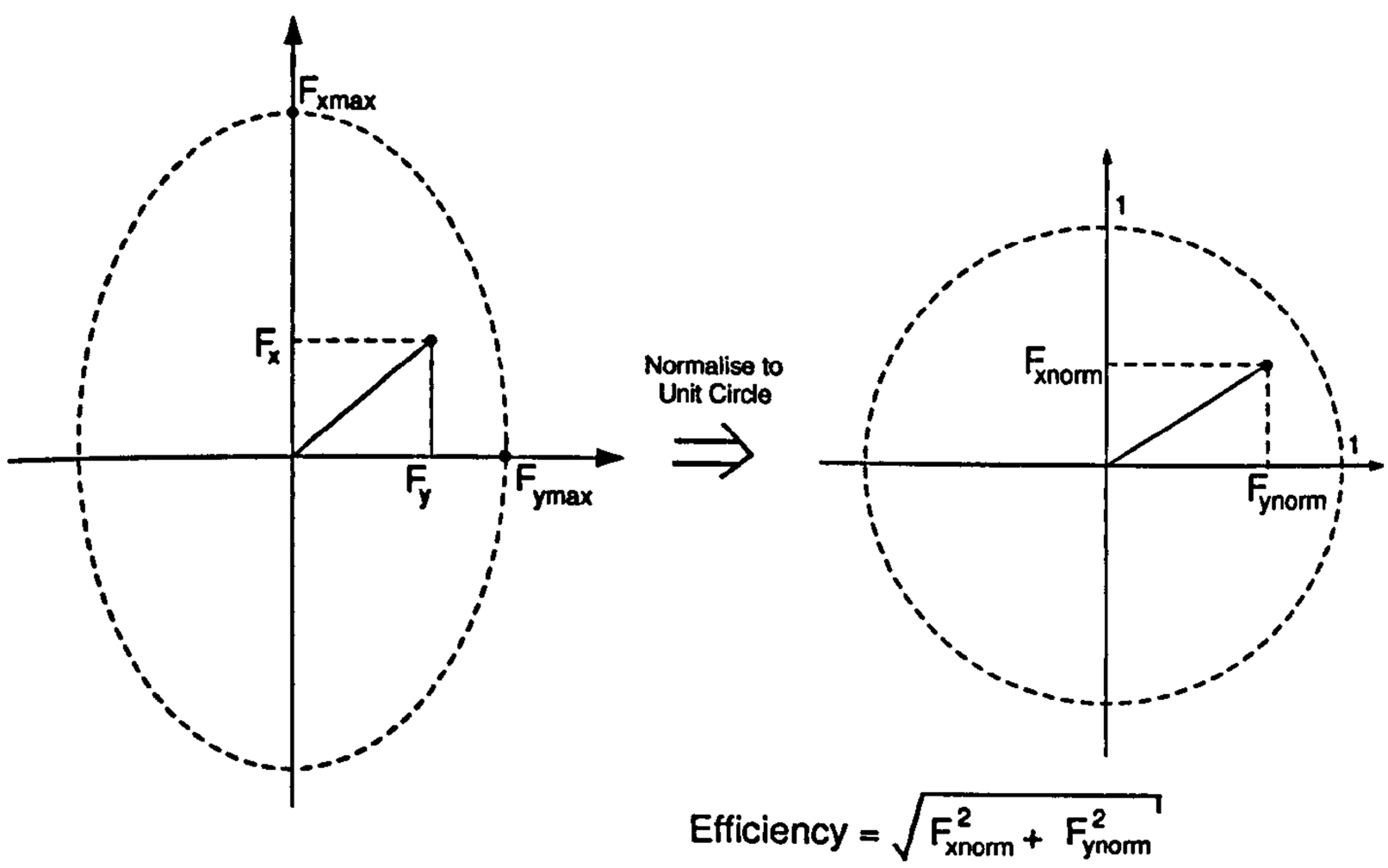


Figure B.3: Computation of tyre efficiency

Appendix C

Non-linear Programming

Many problems in engineering design can be summarised as to design a system that satisfies a set of requirements whilst achieving the goal of minimising a cost function. In many cases these problems can be approached by constructing a mathematical model of the problem then applying numerical optimization methods to the model. The model will typically have a number of parameters and we wish to find suitable values for the parameters to achieve the goal. The goal can be represented as a scalar valued function $f(\bar{x})$ where $\bar{x}$ is the n dimensional vector of parameters usualy termed the independent variables. We are only concerned here with continuous problems where f and $\bar{x}$ are real numbers. The problem can be summarised as to follows:

Minimize:

$$f(\bar{x}) \quad \bar{x} \in R^{n_x} \tag{C.1}$$

Subject to:

$$c_i(\bar{x}) \quad = 0 \quad \ldots \; i \in [1, 2, \ldots p] \tag{C.2}$$

(C.3)

$$c_j(\bar{x}) \quad \leq 0 \quad \ldots \; j \in [1, 2, \ldots q] \tag{C.4}$$

(C.5)

The function $f(\bar{x})$ is known as the objective function. Equations C.3 and C.5 define the equality and inequality constraints. In the category of problems we are interested in, the objective and constraint functions are computationaly complex e.g. involving the numerical solution of a non-linear ODE boundary value problem. We also assume that n is sufficiently large (typically > 50) so that simple visualization of the functions using 2 and 3 dimensional plots is not feasible. The problem stated above is a Non-Linear Program and the methods associated with its solution are known as Non-Linear Programming. Practical software implementations of algorithms for solving these problems are generally know as NLP codes.

C.1 Local and Global Minima, Convexity of the objective and feasible region

We begin by defining local and global minimising points. A function $f(\bar{x})$ has a minimum at a point $\bar{x}^*$ if :

$$f(\bar{x}^*) \leq f(\bar{x}) \text{ when } |\bar{x} - \bar{x}^*| \leq \epsilon \text{ for some value } \epsilon > 0 \tag{C.6}$$

This minimum is also a global minimum if :

$$f(\bar{x}^*) \leq f(\bar{x}) \forall \bar{x} \tag{C.7}$$

Points satisfying equation C.6 but not C.7 are local minima. Figure C.1 shows two example functions. On the interval shown C.1(a) has a single minimum and C.1(b) has three minima. If $f(\bar{x})$ is a convex function then it posesses a single global minimum. Hence

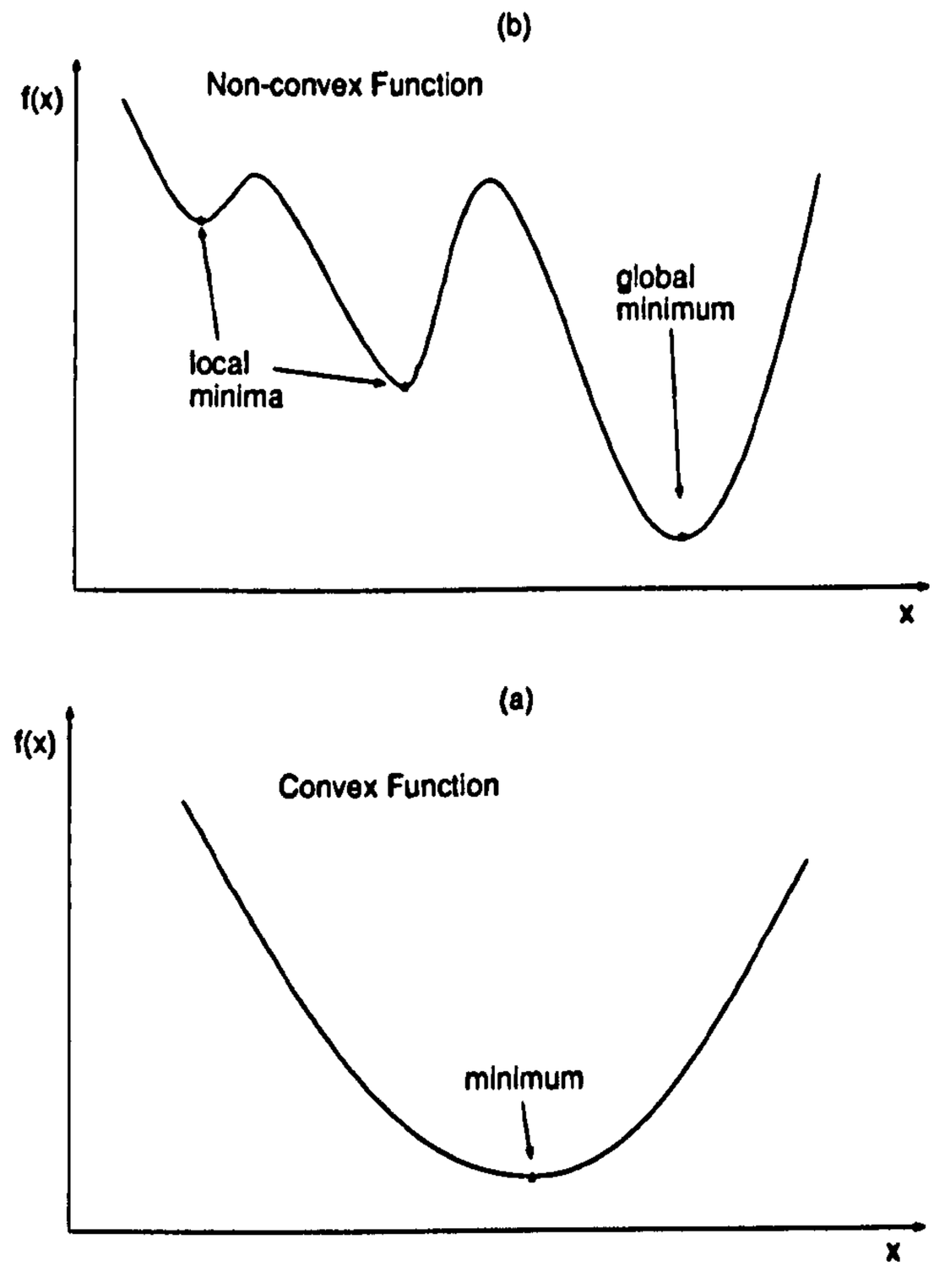


Figure C.1: Convex and non Convex functions

unconstrained problems with convex objectives posess a single minimum. A continuously differentiable function $f(\bar{x})$ is convex on an interval C if:

$$\forall\ \bar{x}_1, \bar{x}_2\ \in C \text{ then } f\left(\frac{\bar{x}_1 + \bar{x}_2}{2}\right) \leq \frac{f(\bar{x}_1) + f(\bar{x}_2)}{2} \tag{C.8}$$

Another useful definition of convexity is achieved by considering how $f(\bar{x})$ divides R^n the vector space it occupies into two sets. The first set contains the points 'above' the function and the second set the points 'below'. If the set of points 'above' the function is convex then $f(\bar{x})$ is convex. A set C is convex if all points on the line connecting any two points $\bar{x}_1, \bar{x}_2 \in C$ are also $\in C$.

In constrained problems we must also consider the set of all points that satisfy all of the constraints. This is known as the feasible set or feasible region. If this set is non-convex then the NLP problem may posess one or more local minima. The formal definition of a constrained minimum point is given later. Figure C.2 shows the contours of an example two dimensional function $f(\bar{x})$ and the line corresponding to the zero value of a single inequality constraint. In figure C.2(a) the objective function and feasible set are convex leading to a single constrained minimising point. In C.2(b) the objective function is convex but the feasible region is not leading to two minimising points.

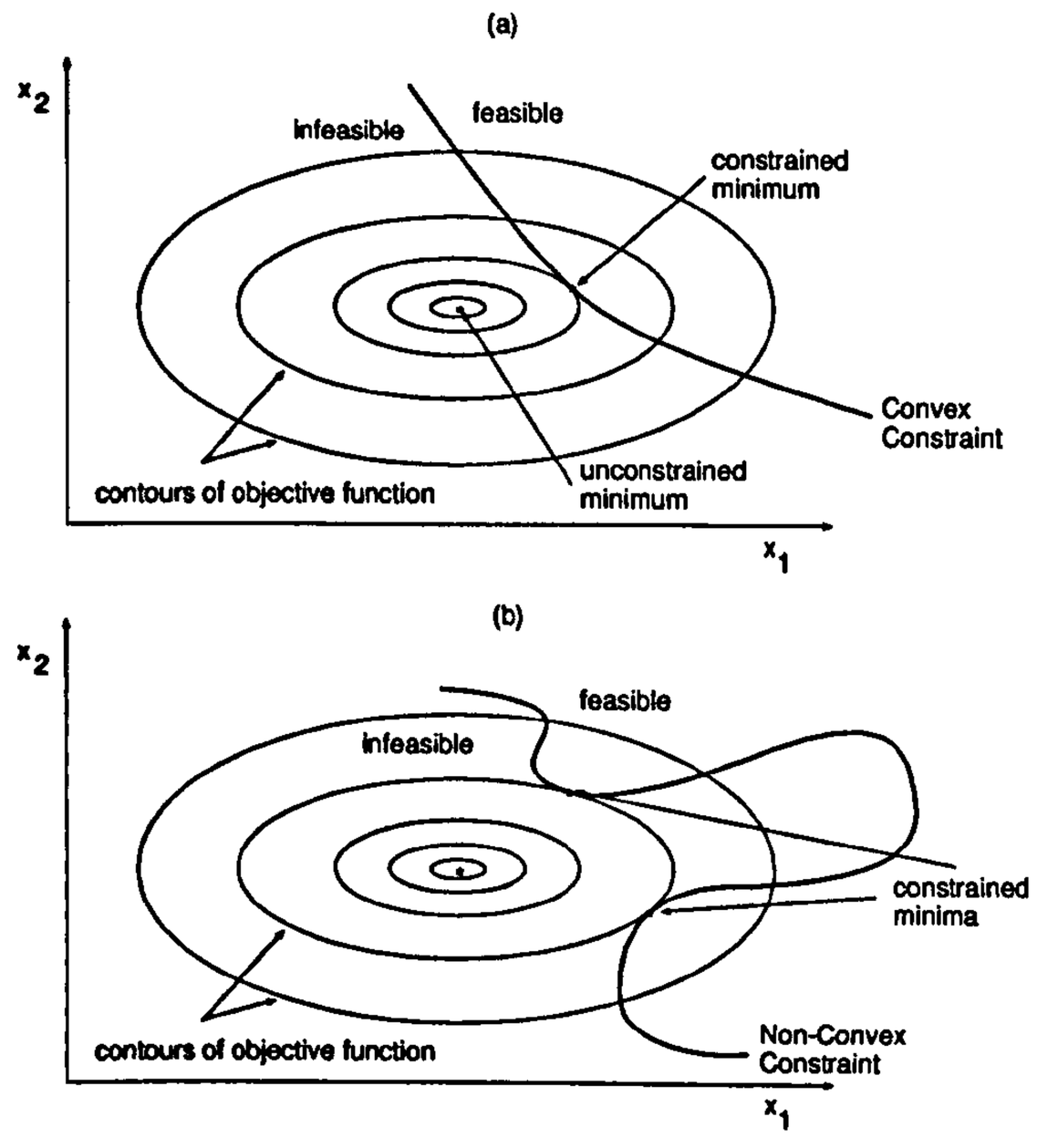


Figure C.2: Convex and non Convex Constraints

C.1.1 Definition of Unconstrained minimum

In the most simple unconstrained case, we wish to find values of the vector $\bar{x}$ corresponding to stationary points of the scalar valued function $f(\bar{x})$. At a stationary point the derivative of the function $f(\bar{x})$ given by $g(\bar{x})$ is equal to zero. For the stationary point to be a minimum we also require that the curvature (second derivative) of the function be positive. Hence a minimising point is a value $\bar{x}^*$ where $g(\bar{x}^*) = 0$ and the matrix of second derivatives $G(\bar{x}^*)$

is positive definite. The matrix $G(\bar{x})$ (defined in equation C.9) is known as the Hessian.

$$G = \begin{bmatrix} \frac{d^2f}{dx_1dx_1} & \frac{d^2f}{dx_1dx_2} & \cdots & \frac{d^2f}{dx_1dx_n} \\ \frac{d^2f}{dx_2dx_1} & \frac{d^2f}{dx_2dx_2} & \cdots & \frac{d^2f}{dx_2dx_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{d^2f}{dx_ndx_1} & \frac{d^2f}{dx_ndx_2} & \cdots & \frac{d^2f}{dx_ndx_n} \end{bmatrix} \tag{C.9}$$

C.1.2 Definition of Constrained Minimum, the Karush Kuhn Tucker (KKT) point

The definition of a constrained minimum is more complex. We will consider first the case for problems with only equality constraints. We start by defining a 'direction' in the n dimensional space from a point $\bar{x}_k$ as the line $\bar{x} = \alpha\bar{s} + \bar{x}_k$ where $\bar{s}$ is an n dimensional vector and α is a scalar. At a constrained minmum point $\bar{x}^*$, there can be no direction in which both the value of $f(\bar{x})$ is reduced and feasibility is maintained. For any direction to maintain feasibility with respect to the equality constraints:

$$\bar{s}^T\bar{a}_i^* = 0 \tag{C.10}$$

Where $\bar{a}_i$ is the vector of gradients for constraint i with respect to $\bar{x}$:

$$\bar{a}_i = \begin{bmatrix} \frac{dc_i}{dx_1} & \frac{dc_i}{dx_2} & \cdots & \frac{dc_i}{dx_n} \end{bmatrix}^T \tag{C.11}$$

For any direction to maintain descent (reduce the value of the objective function):

$$\bar{s}^T\bar{g}^* < 0 \tag{C.12}$$

For equations C.10 and C.12 to hold then $\bar{g}^*$ must be a linear combination of the vectors $\bar{a}_i^*$ as shown in equation C.13. The matrix A is the matrix whose columns are made up of the constraint gradient vectors a_i (equation C.15. The multipliers λ_i^* are refered to as Lagrange Multipliers. A formal proof of this statement is given in [46].

$$\bar{g}^* = A^*\lambda^* \tag{C.13}$$

$$c_i(\bar{x}) = 0 \tag{C.14}$$

$$A = \begin{bmatrix} \bar{a}_1 & \bar{a}_2 & \ldots & \bar{a}_p \end{bmatrix} \tag{C.15}$$

The problem of finding a constrained minimum consists of finding vectors $\bar{\lambda}^*$ and $\bar{x}^*$ that satisfy equations C.13 and C.14. These results can be re-stated using the Lagrangian function:

$$L(\bar{x}, \bar{\lambda}) = f(\bar{x}) - \sum \lambda_i c_i(\bar{x}) \tag{C.16}$$

The conditions satisfying equations C.13 and C.14 can be re-stated as:

$$\frac{dL(\bar{x}, \bar{\lambda})}{d\bar{x}} = 0 \tag{C.17}$$

$$\frac{dL(\bar{x},\bar{\lambda})}{d\bar{\lambda}} = 0 \tag{C.18}$$

When inequality constraints are added to the problem we consider them to form two groups. Those which are active (zero) at the minimising point and those which are not active but satisfied (less than zero). The active inequalities are effectively equality constraints. A constrained minimising point $\bar{x}^*$ is defined by the Karush-Kuhn-Tucker (KKT) conditions as follows:

$$c_j(\bar{x} \leq 0 \tag{C.19}$$

$$\lambda_j \leq 0 \tag{C.20}$$

$$\lambda_i c_i(\bar{x}) = 0 \tag{C.21}$$

$$\lambda_j c_j(\bar{x}) = 0 \tag{C.22}$$

In order to satisfy the second order condition that curvature is positive at the solution, the Hessian of the Lagrangian at $\bar{x}^*$ must be positive semi-definite.

C.2 Locating a minimum

Practical approaches to locating a minimising point $\bar{x}^*$ involve generating a series of iterates from an initial guess: $\bar{x}_k, \bar{x}_{k+1}, \bar{x}_{k+2} ... \bar{x}^*$. We will consider first the unconstrained case. Each successive point is typically generated as a minimising point on the line defined by the search direction s. The simplest approach is that of 'steepest' descent. Here the search direction is defined as the negative of the objective gradient as shown in equation C.23.

$$s_k = -\dot{f}(\bar{x}_k) \tag{C.23}$$

The process of locating the minimising point is usually termed the 'line search' sub-problem. The steepest descent method can be considered as fitting a linear model to $f(\bar{x})$. Where $f(\bar{x})$ is not linear and in particular where the ratio of the largest and smallest values in $\dot{f}(\bar{x})$ is large, the steepest descent method makes very slow progress towards a minimum. An alternative approach is based around fitting a quadratic to $f(\bar{x})$ around x_k as shown in equation C.24.

$$q(\bar{x}) = \frac{1}{2}(\bar{x} - \bar{x}_k)^T G(\bar{x} - \bar{x}_k) + \bar{b}^T(\bar{x} - \bar{x}_k) + \bar{c} \tag{C.24}$$

The line search direction is then computed according to Newtons method as shown in equations C.25 and C.26 where G is the Hessian matrix and H is its inverse.

$$s = -H\dot{f} \tag{C.25}$$

$$H = G^{-1} \tag{C.26}$$

In the type of problem we are interested in here, second order derivative information is not directly available and estimation of the Hessian by finite differences would be very expensive in terms of computation time. Methods known as 'Quasi-Newton' use a search direction as in equation C.25 but use and estimate G or H. The estimates are formed based on the differences between the first order gradients at successive iterates. Various approaches to Hessian approximation are given in [46].

In practice, quadratic models form workable local approximations to a wude range of functions. However, the quality of fit of the quadratic model will reduce as the distance from $\bar{x}_k$ increases. In order to prevent large steps being taken on the basis of a poor search direction it is common practice to restrict the radius of the line search. Such methods are typically termed 'restricted step' or 'trust region'.

Quasi-Newton methods are also applicable to constrained problems. For constrained problems we are interested in finding stationary points of the Lagrangian. This achieved by computing the search direction based on the gradients and estimated Hessian of the Lagrangian function. The added issue for constrained problems is that the Lagrange multipliers must also be estimated at each iterate.

We summarise below the outline computational steps of a Quasi-Newton method. This conists of 'Major' steps at which the search direction is computed and minor steps at each point along the line search where the objetive and gradient are computed.

- Estimate the gradients of the objective function and constraints at the current point
- Check if the current point is a suitable minimum, if so, quit
- Estimate Lagrange multipliers at the current point
- Update the Hessian of the Lagrangian at the current point
- Choose a search direction
- Search along this direction for a minimum point satisfying the constraints to a suitable tolerance
- Make this minimum the current point

C.2.1 'Interior Point' or 'Feasible' Methods

In many practical optimization problems encountered in engineering design, the objective function may be undefined or discontinuous for points outside of a given region. In these cases, it is important that the sequence of iterates are all contained in the set for which the objective function is suitably defined. Some optimization algorithms ('Interior Point' or 'Feasible' methods) generate a sequence of iterates all of which satisfy the inequality constraints. In practice constraint functions can be added to the problem to define this region.

C.2.2 Problematic Objective and Constraint Functions

NLP codes can run into practical difficulties for various reasons. Figure C.3 is an example of a problematic objective function. It consists of a deep, steep sided, narrow and curved valley. The gradients of the sides of the valley are high but the gradient along the middle of the valley floor is very low. This spread of gradients can cause numerical difficulties. When computing a search direction as shown in equation C.25, if the ratio of the largest and smallest elements in g approaches the machine precision then the smallest gradient may have little influence on the search direction. Under these circusmtances, the series of iterates produced may simply oscillate up and down the valley sides making little or no progress towards a minimum. Another source of problems is caused when the objective

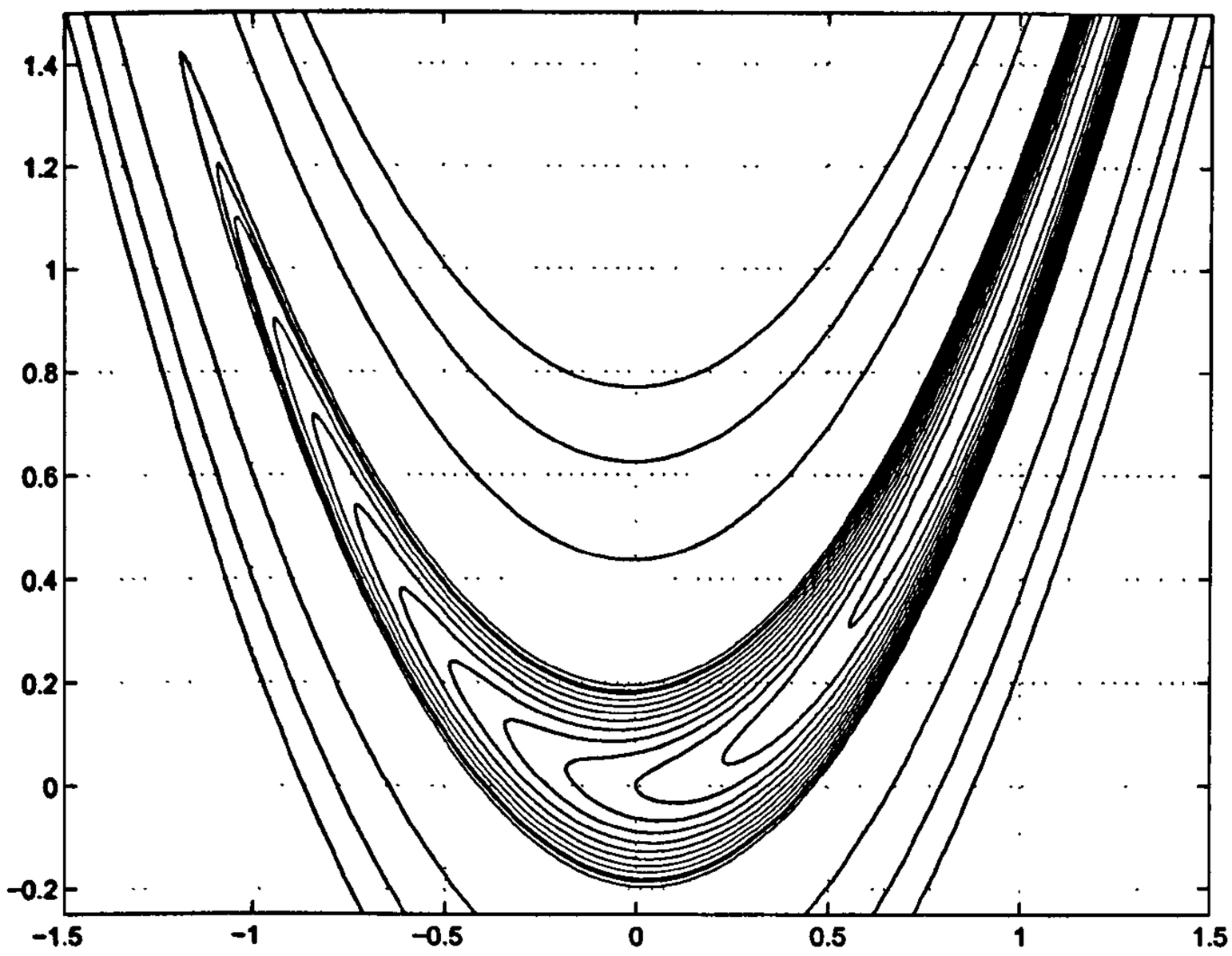


Figure C.3: Contours of Rosenbrock's Banana function at $f(x)$= 0.2, 0.6, 1.0, 1.4, 1.8, 2.2, 2.6, 3.0, 3.4, 3.8, 4.2, 4.4, 4.8, 20, 40, 60

or constraint functions are highly non-quadratic or in the worst case discontinuous. This will typically result in the change of $\bar{x}$ during each line search being very small as the objective and / or constraints rapidly diverge from quadratic behaviour. This results in slow convergence towards the minimum.

C.2.3 Alternative Approaches and Global optimization

The methods discussed so far are designed to locate local minima (or the global minimum in problems with non local mininma). In general we are interested in finding the global minimum but in practice it is diffcult to state with certainty that a given solution is globally minimal. We are typically dealing with a complex non-linear function that is effectively a 'black box' model. That is to say it is too complex for us to understand. The only way a

globally minimal solution can be guaranteed is by exhaustive search of the solution space and this is clearly impractical. A commonly adopted method is to solve the problem from several different start points and study the 'route' taken to the solution for coverage of the vector space. If each of the independent variables has been exercised over a wide range and if the same minimising point is located from any start point. Confidence that the point is globally minimising can be increased. Various alternative approaches exist to the solution of the NLP problem motivated by the desire to search more of the solution space . 'Simulated annealing' is based on producing a series of iterates on an essentialy random basis. The algorithm has associated with it a 'temperature' which gradually reduces during the course of the solution. At each step, a trial iterate is formed by adding a random vetor to the current value of $\bar{x}$. If the objective is reduced the trial point is typically taken as the next iterate. If it is increased the trial iterate may also be selected as the new iterate on a statistical basis. Otherwise the trial is rejected and a new trial is generated. At high temperatures the magnitude of the random increments to $\bar{x}$ are larger and the probability of accepting a trial that results in an increase in the objective is higher. As the temperature reduces the random steps become smaller and the willingness to acceptance an increase in the objective is reduced. This method was used by the Author with some success during the early stages of this work. It is however very slow to converge.

A further alternative is the genetic algorithm. This is designed to emulate the natural selection process in organisms. A 'population' of candidate solutions is maintained which are in someway 'mutated' then split into partial solution vectors before being spliced. This produces a population of offspring from which the best are selected for a further round of breeding. The algorithm is designed to search a large part of the solution space.

C.3 CFSQP

The NLP code utilised by the MTM method developed in this thesis is the Feasible-Sequential-Quadratic-Programming (FSQP) code in its C language implementation (CFSQP). The algorithm and code was developed by Craig Lawrence, Jian L. Zhou and Andre L. Tits at the Institute for Systems Research, University of Maryland [10]. It is available commercially from AEM design www.aemdesign.com. CFSQP is well suited to the MTM problem for the following reasons:

- It is an interior point method. The objective function is only evaluated at points satisfying all inequality constraints. Hence highly non-linear regions of the objective can be excluded from the space that is sampled en-route to the solution.

- As we show in section 3.3.3 the MTM problem as developed in this thesis is non-smooth. NLP codes differ in their tolerance of this non-smoothness. CFSQP is quite well suited to non-smooth problems due at least in part to the simplicity of its line search procedure.

- Non-monotone line search. When solving non-linear constrained problems we must proceed along curved constraints so a line is not the best choice.

Appendix D

Quasi-Steady-State Lap time simulation

Quasi Steady state (QSS) lap time simulation codes have been in succesful widespread use in the motor racing industry for more than 20 years. Due to the general level of secrecy in the industry and the commercial nature of the available codes, very little has been published on the subject of the detailed implementation of steady state lap time simulators. We present here a simple algorithm for a QSS lap time simulator that captures only the essential features of the approach.

A general vehicle model is given by the differential equation D.1. Table D.1 describes the state vector $\bar{m}$ and its derivatives for the QSS model described in Appendix A.

$$\frac{d(\bar{m}(t))}{dt} = \bar{F}(\bar{m}(t), \bar{u}(t)) \tag{D.1}$$

The SAE definition of steady state is as follows:

> Steady State exists when periodic (or constant) vehicle responses to periodic (or constant) control and/or disturbance inputs do not change over an arbitrarily long time. The motion responses in steady-state are referred to as steady-state responses

State Vector Element	Derivate
v_x, longitudinal velocity m/s	$\dot{v}_x$ or A_x, longitudinal acceleration
v_y, lateral velocity m/s	$\dot{v}_y$, lateral velocity acceleration
r, yaw rate m/s	$\dot{r}$, yaw acceleration
ω_{LF}, LF wheel speed (rads/s)	$\dot{\omega}_{LF}$ LF wheel acceleration
ω_{RF}, RF wheel speed (rads/s)	$\dot{\omega}_{RF}$ RF wheel acceleration
ω_{LR}, LR wheel speed (rads/s)	$\dot{\omega}_{LR}$ LR wheel acceleration
ω_{RR}, RR wheel speed (rads/s)	$\dot{\omega}_{RR}$ RR wheel acceleration

Table D.1: Test Configuration

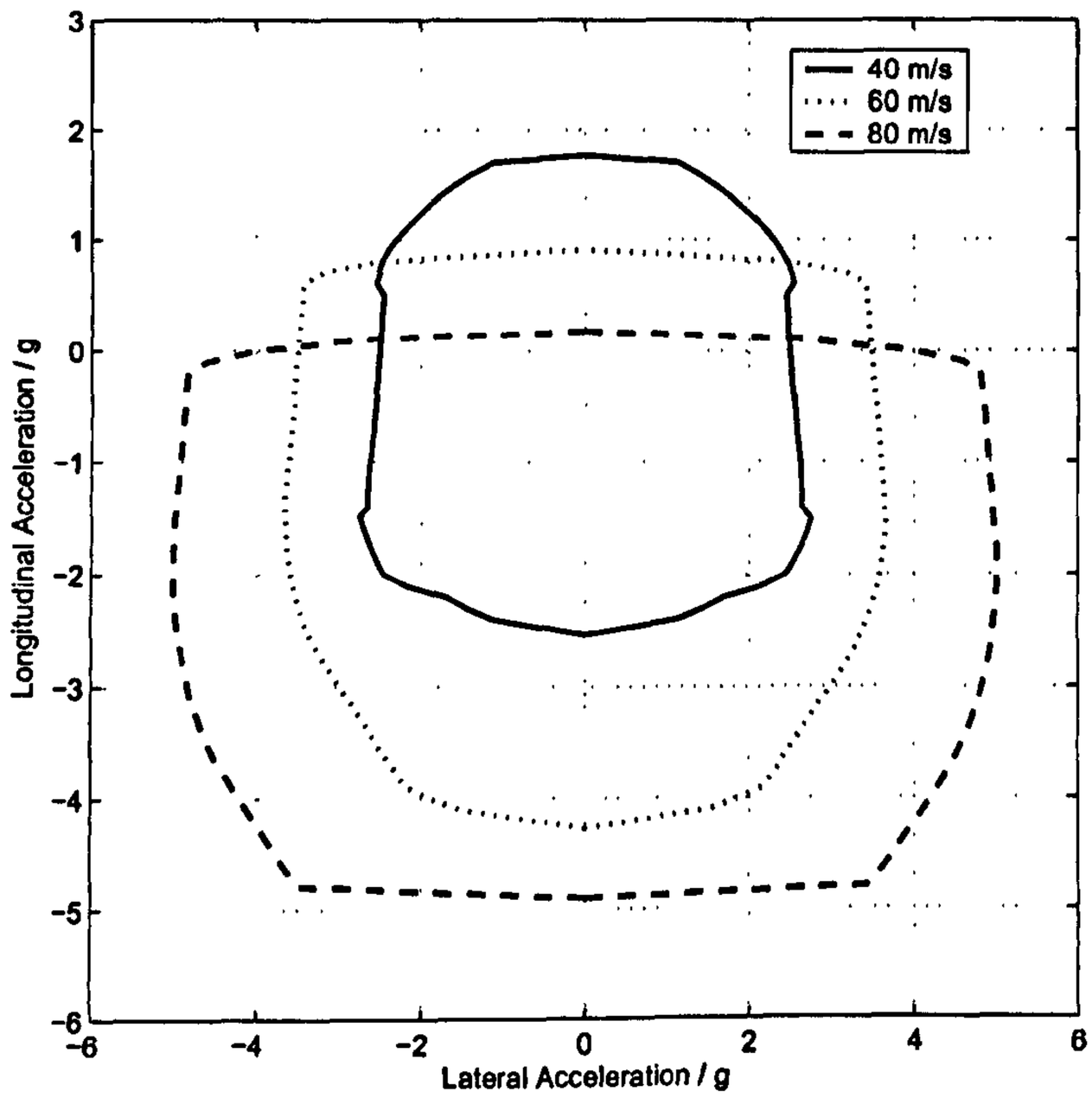


Figure D.1: gg diagram for QSS car set-up A with tyre set A

Appendix E

Vehicle Stability Derivatives

E.1 Stability Derivatives

The following derivatves describe the rate of change of vehicle yaw moment N and total lateral force F_y with yaw rate, body side slip and steer angle [36].

E.1.1 Yaw Stiffness N_β

The yaw stiffness is given by the ratio of the change in yawing moment N resulting from a small change in chassis side-slip angle β as follows:

$$N_\beta = \frac{\Delta N}{\Delta \beta} \tag{E.1}$$

If the change in N is such as to attempt to reduce the change in β then N_β is positive. If N_β is negative, an increase in β will result in a change in yaw moment that tends to promote a futher increase in β. Under these circumstances the vehicle is considered to be in an 'open loop unstable' state. That is, without intervention from the driver, β will continue to increase until N_β becomes positive again. This process is what typically happens when a car 'spins', the car rotates through 180° and becomes stable again travelling backwards.

E.1.2 Yaw Damping N_r

The yaw damping is given by the ratio of the change in yawing moment N resulting from a small change in yaw rate r as follows:

$$N_r = \frac{\Delta N}{\Delta r} \tag{E.2}$$

If the change in N is such as to attempt to reduce the change in r then N_r is negative.

E.1.3 Yaw Control Moment Derivative N_δ

The control moment is given by the ratio of the change in yawing moment N resulting from a small change in steer angle δ as follows:

$$N_\delta = \frac{\Delta N}{\Delta \delta} \tag{E.3}$$

If the change in N is such as to attempt to reduce the change in δ then N_δ is negative.

E.1.4 Damping in Sideslip Y_β

This derivative is the ratio of the change in total lateral force resulting from a small change in β as follows:

$$Y_\beta = \frac{\Delta F_y}{\Delta \beta} \tag{E.4}$$

E.1.5 Lateral Force / Yaw Rate Coupling Derivative Y_r

This derivative is the ratio of the change in total lateral force resulting from a small change in yaw rate r as follows:

$$Y_r = \frac{\Delta F_y}{\Delta r} \tag{E.5}$$

E.1.5.1 Control Force Derivative Y_δ

This derivative is the ratio of the change in total lateral force resulting from a small change in steer angle δ as follows:

$$Y_\delta = \frac{\Delta F_y}{\Delta \delta} \tag{E.6}$$

E.2 Understeer Angle

The understeer angle is a crude but commonly used method of categorizing vehicle limit behaviour. The low speed or 'kinematic' steer angle for a vehicle with wheelbase L is defined as follows:

$$\delta_k = \frac{Lr}{v_x} \tag{E.7}$$

The understeer angle is the difference between the steer angle and the kinematic steer angle as follows:

$$\theta_{uang} = \delta - \delta_k \tag{E.8}$$

The sign convention is such that θ_{uang} is positive when δ exceeds δ_k (understeer).

Appendix F

Software Implementation - LORA - Laptime optimization for Racecar Analysis

A significant part of the effort involved in producing this thesis consisted of the development of a practical software tool to evaluate and analyse the MTM algorithms and to produce the presented results. A brief summary of the of the role of the software is as follows:

- Implement the MTM algorithms
- Implement the computational procedures as described in section 2.15
- Implement the vehicle and tyre models and associated ODE solver
- Implement representations of the race track, including bumps and kerbs
- To provide a graphical user interface to allow visualisation of the MTM algorithm as it operates

The software has been developed under the name LORA (Laptime optimization for Racecar Analysis). It was developed in the C++ language with a user interface using the Microsoft Foundation Classes (MFC). The software was designed and developed in an Object Oriented (OO) style. This appendix gives a brief description of the software.

F.1 Software Outline

Each class is derived from a common base class that supports methods for persistent storage and retrieval to/from files. Each class implements a method to store itself in an XML formatted file. The file includes class version data to support backwards compatible operation with previous versions of the software. The same methods allow a class to transfer itself over an IP socket to allow classes to be distributed across processor networks.

F.1.1 Optimizer

In the early stages of the research, a number of NLP codes were evaluated. In order to provide a simple common interface to the application code regardless of the particular NLP code used, an optimizer interface class was defined. This interface defines the essential features common to all NLP codes. It includes a virtual function definition for the method used to calculate the objective and constraints.

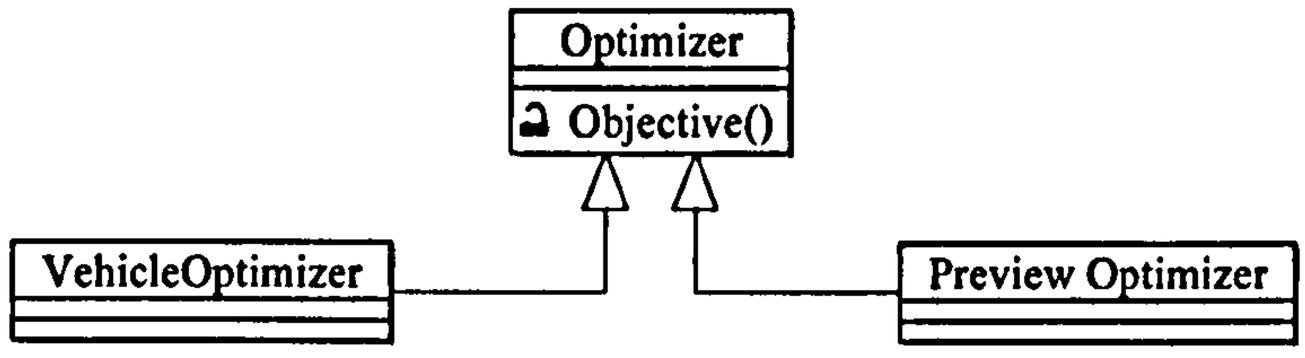


Figure F.1: Optimizer Class Diagram

Classes implementing the distance and time MTM methods and the QSS simulator use the optimizer class to implement their respective algorithms.

F.1.2 Vehicle and Tyre Models

Figure F.2 shows a subset of a class diagram for the vehicle and tyre model classes. The vehicle and tyre are each represented by an interface class. This allows further vehicle and tyre models to be easily added to the software. The autosim generated code for the sprung car is encapsulated into a wrapper class.

F.1.3 Experiment

The experiment class is responsible for implementing the computational procedures outlined in section 2.15. An experiment consists of the following:

- A Vehicle and tyre set
- A Racetrack
- A computational procedure
- The experiment configuration (vehicle parameter values to sweep, MTM algorithm parameters to sweep etc.)

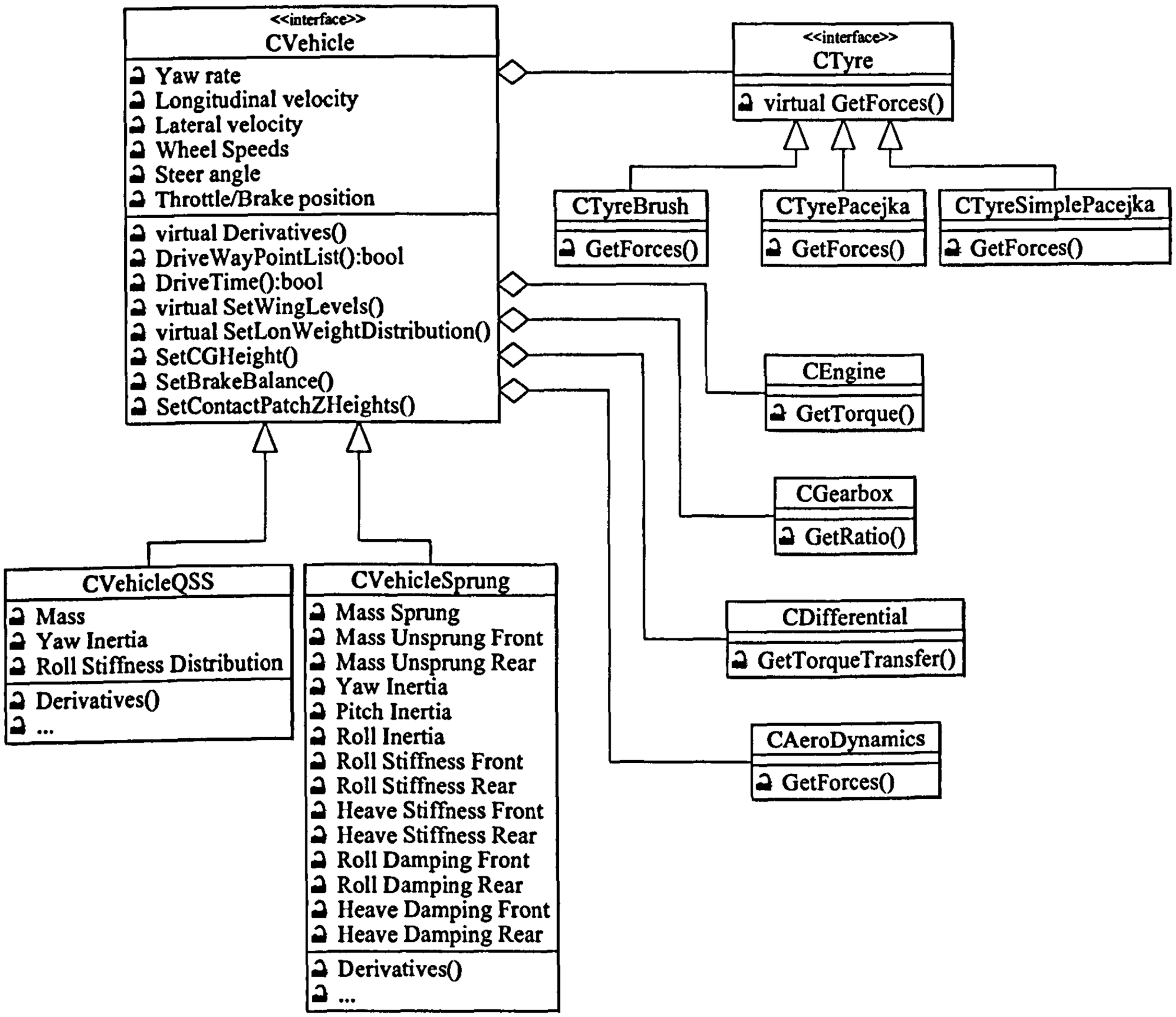


Figure F.2: Vehicle, Tyre and Support Classes

1

Bibliography

[1] Snyman J.A. Hay A.M. The dynamic-q optimization method: An alternative to sqp ? *Computers and Mathematics with Applications, Vol 44, pp 1589-1598*, 2002. [cited at p. 176]

[2] Tan H.Z. Srinivasan M.A. Eberman B. Cheng B. Human factors for the design of force-reflecting haptic interfaces. *DSC-Vol.55-1, Dynamic Systems and Control (The American Society of Mechanical Engineers)*, 1994. [cited at p. 59]

[3] John Bellingham. Receding horizon control of autonomous aerial vehicles. *ACC*, 2002. [cited at p. 101]

[4] M.F. Brayshaw, D.L. Harrison. Use of numerical optimization to determine the effect of the car roll stiffness distribution on race car performance. *Proceedings of the IMECHE Part D Journal of Automobile Engineering*, 2005. [cited at p. 10]

[5] L.E. Browne, A.L. Wickliffe. Rubber emmisivity and thermal state of tires. *Tire Science and Technology, Vol. 7*, 1979. [cited at p. 151]

[6] L.E. Browne, A.L. Wickliffe. Parametric study of convective heat transfer coefficients at the tire surface. *Tire Science and Technology, Vol. 8*, 1980. [cited at p. 152]

[7] von Stryk O Butz T. Optimal control based modeling of vehicle driver properties. *SAE World Congress Detroit Michigan 2005-01-0420*, 2005. [cited at p. 60]

[8] Daniele Casanova. *On Minimum Time Manoeuvring: The theoretical optimal lap.* PhD thesis, Cranfield University, 2000. [cited at p. 1, 13, 15, 23, 25, 33, 72, 174, 185, 188, 214]

[9] G. et.al. Corliss. *Automatic differentiation of algorithms: from simulation to optimization.* Springer, 2000. [cited at p. 14]

[10] Andre L.Tits Craig Lawrence, Jian L. Zhou. Users guide for cfsqp version 2.5: A c code for solving (large scale) constrained nonlinear (minimax) optimization problems, generating iterates satisfying all inequality constraints. *Manual*, 1997. [cited at p. 29, 212]

[11] Brayshaw D. *The use of numerical optimisation to determine on-limit handling behaviour of race cars.* PhD thesis, Cranfield University, 2004. [cited at p. 215]

[12] R. S. Sharp D. Casanova and P. Symonds. Minimum time manoeuvring: The significance of yaw inertia. *Vehicle System Dynamics*, 34, 2000. [cited at p. 13]

[13] R. S. Sharp D. Casanova and P. Symonds. On the optimisation of the longitudinal location of the mass centre of a formula one car for two circuits. *AVEC 02*, 2002. [cited at p. 13]

[14] Kane T.R. Levinson D.A. *Dynamics: Theory and Applications.* McGraw-Hill, 1985. [cited at p. 119]

[15] Siegler B. Crolla D.A. Lap time simulation for racing car design. *SAE 2002-01-0567*, 2002. [cited at p. 15]

[16] Siegler B. Deakin A. Crolla D.A. Lap time simulation: Comparison of steady state, quasi-static and transient racing car cornering strategies. *SAE 2000-01-3563*, 2000. [cited at p. 15]

[17] J.C. Dixon. *Tires, Suspension and Handling.* SAE, 1996. [cited at p. 121, 122, 124, 140]

[18] Williams M.L. Landel R.E. Ferry F.D. The temperature dependence of relaxation mechanisms in amorphous polymers and other glass-forming liquids. *J. Am. Chem. Soc. Vol. 77(14) pp 3701-3707*, 1955. [cited at p. 145, 146]

[19] Saccon A Frezza R, Beghi A. Model predictive control for path following with motorcycles: Application in the development of the pilot model for virtual prototyping. *IEEE Conference on Decision and Control*, 2004. [cited at p. 60]

[20] Giancarlo Genta. *Motor Vehicle Dynamics: Modelling and Simulation.* World Scientific, 1997. [cited at p. 140]

[21] Saunders M.A. Gill P.E., Murray W. Snopt: An sqp algorithm for large-scale constrained optimization. *SIAM Journal on Optimization*, 12(4):979–1006, 2002. [cited at p. 14, 172]

[22] Wright M.H. Gill P.E, Murray W. *Practical Optimization.* Academic Press, 1986. [cited at p. 31]

[23] J.M. Golden. A molecular theory of adhesive rubber friction. *J. Phys. A: Math. Gen., Vol. 8, No, 6*, 1975. [cited at p. 145]

[24] Van Loan C.F. Golub G.H. *Matrix Computations.* Johns Hopkins University Press, 1996. [cited at p. 30, 31]

[25] Best M.C Gordon T.J. A sequential dual model approach to lap optimisation. *Proceedings of the 6th International Symposium on Advanced Vehicle Control*, 2002. [cited at p. 15]

[26] Best M.C Gordon T.J. Simultaneous optimisation of vehicle parameter and control to examine validity of handling control assumptions. *Proceedings of the 6th International Symposium on Advanced Vehicle Control*, 2002. [cited at p. 15]

[27] Fujita S Goto N, Chatani K. H infinity model of the human pilot controlling unstable aircraft. *IEEE*, 1995. [cited at p. 59]

[28] Corliss G.F. Griewank A. *Automatic Differentiation of Algorithms: Theory, Implementation and Application*. Springer-Verlag, New York, 2002. [cited at p. 14]

[29] K.A. Grosch. The relation between the friction and visco-elastic properties of rubber. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences, Vol. 274, No. 1356. pp. 21-39*, 1963. [cited at p. 145, 146]

[30] Pacejka H.B. *Tyre and Vehicle Dynamics*. Elsevier, 2002. [cited at p. 139, 140, 142, 143, 144, 197]

[31] Yavari B. Tworzydlo W.W. Bass J.M. A thermomechanical model to predict the temperature distribution of steady state rolling tires. *Tire Science and Technology, Vol. 21, pp 163-178*, 1993. [cited at p. 142, 150]

[32] Arthur E. Bryson Jr. *Dynamic Optimization*. Addison-Wesley, 1999. [cited at p. 11, 29]

[33] Best M.C Komatsu A, Gordon T.J. Vehicle path optimisation using a time-variant linear optimal reference control. *Proceedings of the 6th International Symposium on Advanced Vehicle Control*, 2002. [cited at p. 15]

[34] The MathWorks. *Matlab Optimization Toolbox*. The MathWorks, www.mathworks.com, 1990. [cited at p. 15]

[35] The MathWorks. *Matlab Symbolic Math Toolbox*. The MathWorks, www.mathworks.com, 1993. [cited at p. 15]

[36] Milliken. *Race Car Vehicle Dynamics*. SAE, 1995. [cited at p. 9, 177, 217]

[37] Els P.S. Uys P.E. Snyman J.A. Thoresson M.J. Gradient-based approximation methods applied to the optimal design of vehicle suspension systems using computational models with severe inherent noise. *Mathematical and Computer Modelling, Vol 43, pp 787-901*, 2006. [cited at p. 176]

[38] Stirling Moss and Lawrence Pomeroy. *Design and Behaviour of the Racing Car*. William Kimber, London, 1963. [cited at p. 56]

[39] Neilson P.D. Speed of response or bandwidth of voluntary system controlling elbow position in intact man. *Medical and Biological Engineering and Computing*, 10(4), 1972. [cited at p. 59, 60]

[40] B.N.J Persson. Theory of rubber friction and contact mechanics. *Journal of Chemical Physics*, 2001. [cited at p. 145, 146]

[41] B.N.J Persson. Rubber friction: role of the flash temperature. *J. Phys: Condensed Matter Vol. 18 pp7789-7823*, 2006. [cited at p. 146, 151]

[42] E Persson, B.N.J. Tosatti. Qualitative theory of rubber friction and wear. *Journal of Chemical Physics*, 2000. [cited at p. 145, 146, 170]

[43] P.Hayney. *The Racing and High-Performance Tyre.* SAE, Warrendale, PA, 203. [cited at p. 145]

[44] Gunther Prokop. Modelling human vehicle driving by model predictive online optimization. *Vehicle System Dynamics*, 35, 2001. [cited at p. 10, 60]

[45] Johnson E.N. Pritchett R. Generic pilot and flight control model for use in simulation studies aiaa 2002-4694. *AIAA Modeling and Simulation Technologies Conference and Exhibit*, 2002. [cited at p. 59]

[46] R.Fletcher. *Practical Methods of Optimization.* Wiley-Interscience, Chichester, 2000. [cited at p. 29, 31, 208, 210]

[47] Smith R.H. On the limits of manual control. *IEEE Transactions on Human Factors in Electronics*, 1963. [cited at p. 59]

[48] Fujikawa T. Funazaki A. Yamazaki S. Tyre tread temperatures in actual contact areas. *Tire Science and Technology, Vol. 22, No. 1*, 1994. [cited at p. 145, 151]

[49] Masaki Yamamoto Shoji Inagakagi, Ikuo Kshiro. Analysis on vehicle stability in critical cornering using the phase-plane method. *AVEC*, 10, 1994. [cited at p. 27]

[50] John T.Betts. *Practical Methods for Optimal Control Using Nonlinear Programming.* SIAM, 2001. [cited at p. 11, 29]

[51] Flannery B.P. Teukolsky S.A, Vetterling W.T. *Numerical Recipes in C.* Cambridge University Press, 1991. [cited at p. 30, 34]

[52] Brooks T.L. Telerobotic response requirements. *Proceedings of IEEE Conference on Systems, Man and Cybernetics*, pages 113–120, 1990. [cited at p. 59]

[53] Peng H Ungoren A.Y. An adaptive lateral preview driver model. *Vehicle System Dynamics. Volume 43, Number 4, pp245-259*, 2005. [cited at p. 60, 101]

[54] P. Van Valkenburgh. *Race Car Engineering and Mechanics.* HPBooks, 1986. [cited at p. 177]

[55] Tsiotras P Velenis E. Minimum time vs maximum exit velocity path optimization during cornering. *Proceedings of the IEEE International Symposium on Industrial Electronics*, 1, 2005. [cited at p. 15]

[56] Tsiotras P Velenis E. Optimal velocity profile for given acceleration limits: Receding horizon implementation. *Proceedings of the American Control Conference*, 3, 2005. [cited at p. 15]

[57] Tsiotras P Velenis E. Optimal velocity profile for given acceleration limits: The half-car model case. *Proceedings of the IEEE International Symposium on Industrial Electronics*, 1, 2005. [cited at p. 15]

[58] Tsiotras P Velenis E. Optimal velocity profile for given acceleration limits: Theoretical analysis. *Proceedings of the American Control Conference*, 2, 2005. [cited at p. 15]

[59] L.E. Whicker D. Browne A.L. Segalman D.J. Wickliffe. Thermomechanical approach to tire power loss modelling. *Tire Science and Technology, Vol. 9*, 1981. [cited at p. 149]